

Spring 6-15-2015

Time-Variant Load Models of Electric Vehicle Chargers

Nicole P. Zimmerman
Portland State University

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Power and Energy Commons](#)

Recommended Citation

Zimmerman, Nicole P, "Time-Variant Load Models of Electric Vehicle Chargers" (2015). *Dissertations and Theses*. Paper 2297.

[10.15760/etd.2294](https://doi.org/10.15760/etd.2294)

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

Time-Variant Load Models of Electric Vehicle Chargers

by

Nicole P. Zimmerman

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Thesis Committee:
Robert Bass, Chair
Richard Campbell
Mark Faust

Portland State University
2015

© 2015 Nicole P. Zimmerman

Abstract

In power distribution system planning, it is essential to understand the impacts that electric vehicles (EVs), and the non-linear, time-variant loading profiles associated with their charging units, may have on power distribution networks. This research presents a design methodology for the creation of both analytical and behavioral models for EV charging units within a VHDL-AMS simulation environment.

Voltage and current data collected from Electric Avenue, located on the Portland State University campus, were used to create harmonic profiles of the EV charging units at the site. From these profiles, generalized models for both single-phase (Level 2) and three-phase (Level 3) EV chargers were created. Further, these models were validated within a larger system context utilizing the IEEE 13-bus distribution test feeder system.

Results from the model's validation are presented for various charger and power system configurations. Finally, an online tool that was created for use by distribution system designers is presented. This tool can aid designers in assessing the impacts that EV chargers have on electrical assets, and assist with the appropriate selection of transformers, conductor ampacities, and protection equipment & settings.

Dedication

For my mother, who never stopped believing in me.

Acknowledgements

I would like to thank my adviser, Dr. Robert Bass, for his guidance and support whenever I sought it and for his ability to allow me to experience failures. I will never forget the important lessons that I learned when I was given the space and time to solve complex problems on my own. I would also like to thank my thesis committee members, including Dr. Bass, Mark Faust, and Dr. Richard Campbell, for lending their valuable time and competencies to review my work and provide me with feedback. A very special thank you to Mike Donnelly, with Mentor Graphics, for sharing his VHDL-AMS expertise and for answering every one of my emails with enthusiasm. Thanks go to my husband-to-be, Michael Woodruff, for never allowing me to give up, acting as a technical sounding board at unusual hours of the night and day, and for giving me the the space and support I needed to finish this work. To Jennifer Carr without whose help, I would not have made it this far; we have much wine to drink together. Finally, I would like to thank our sponsors at PGE for their generous financial support of the Power Engineering Education Laboratory and all of the student researchers in it.

Contents

Abstract	i
Dedication	ii
Acknowledgements	iii
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Background	2
1.1.1 Transformers	3
1.1.2 Power Cables	4
1.1.3 Relays, Switch Gear, and Metering Equipment	5
1.1.4 Capacitors	5
1.1.5 Total Harmonic Distortion	6
1.1.5.1 Total Demand Distortion	6
1.1.6 Triplen Currents	7
1.1.7 System Imbalance	8
1.1.7.1 Power Resolution Tree	8
1.2 EV Charger Models	10
2 Data Collection	12
2.1 Measurement Procedure	12
3 Design Methodology	15
3.1 Models and Schematics	15
3.1.1 EV Charger Components	16
3.1.1.1 Analytical Solution	17
3.1.1.2 Behavioral Solution	20
3.1.2 Three-Phase Test System	20
3.1.2.1 Infinite Three-Phase Source	20
3.1.2.2 Three-Phase Transmission Line	21
3.1.2.3 Delta-Wye Transformer	21

3.1.2.4	Delta-Delta Transformer	22
3.1.2.5	Single-Phase Center-Tapped Transformer	22
3.1.2.6	Cable	22
3.1.2.7	Schematic Design	23
3.1.3	IEEE 13 Node Test System	25
3.1.3.1	Voltage Source	25
3.1.3.2	Single-Phase Transformer	26
3.1.3.3	Voltage Regulator	26
3.1.3.4	Lines	27
3.1.3.5	Shunt Capacitors	27
3.1.3.6	Constant Power Loads	27
3.1.3.7	Constant Current Loads	28
3.1.3.8	Constant Impedance Loads	28
3.1.3.9	Schematic Design	28
4	Results & Analysis	30
4.1	Simulations Analyzed	30
4.1.1	EV Chargers on Three-Phase Systems	30
4.1.1.1	Level 2 Charger	30
4.1.1.2	Level 3 Charger	35
4.1.2	Validation of the 13 Node System	38
4.1.3	EV Chargers on the 13 Node System	41
5	Web-Based Tool	43
5.1	Components	44
5.1.1	Level 2 Charger	44
5.1.2	Level 3 Charger	45
5.1.3	Infinite Three-Phase Source	45
5.1.4	Three-Phase Transmission Line	46
5.1.5	Delta-Wye Transformer	46
5.1.6	Delta-Delta Transformer	47
5.1.7	Single-Phase Center-Tapped Transformer	47
5.1.8	Cable	48
5.2	Capabilities	48
6	Discussion	50
7	Conclusion	52
	Bibliography	53
	Appendix A: Figures	58
A.1	Level 3 Charger Schematic	58

Appendix B: VHDL-AMS Code	59
B.1 Level 2 Charger	59
B.2 Level 3 Charger	69
B.3 Infinite Three-Phase Source	78
B.4 Three-Phase Transmission Line	84
B.5 Delta-Wye Transformer	90
B.6 Delta-Delta Transformer	97
B.7 Single-Phase Center Tapped Transformer	104
B.8 Cable	109
B.9 Voltage Source	116
B.10 Single-Phase Transformer	120
B.11 Voltage Regulator	123
B.12 Resistor	127
B.13 Inductor	129
B.14 Capacitor	131
B.15 Constant Power Load	133
B.16 Constant Current Load	136

List of Tables

4.1	Summary of IEEE 13 node voltage profiles compared to VHDL-AMS system voltages.	39
4.2	Summary of IEEE 13 node phase angles compared to VHDL-AMS system angles.	39
4.3	Summary of IEEE 13 node current flow as compared to VHDL-AMS system angles.	40
4.4	TDD for 13 node system with EV charger loads.	42

List of Figures

1.1	IEEE Standard 1459-2010 powers tree.	9
2.1	Existing and temporary metering at Electric Avenue.	14
3.1	Simplified version of AC harmonic load model.	18
3.2	Time-domain current response to applied sinusoidal voltage.	19
3.3	Radial distribution test system with multiple EV chargers.	24
3.4	The IEEE 13 node radial test distribution system.	29
4.1	Charger currents for Level 2 charger as reproduced in MATLAB.	31
4.2	Charger currents for Level 2 as simulated in SystemVision.	32
4.3	FFT analysis of Level 2 chargers at each of the five charging states.	33
4.4	FFT analysis of Level 2 chargers at 1 st , 3 rd , and 5 th harmonics for each of the five charging states.	34
4.5	Charger currents for Level 3 charger as reproduced in MATLAB.	36
4.6	Charger currents for Level 3 as simulated in SystemVision.	37
4.7	One-line diagram for IEEE 13 node system.	41
A.1	Schematic of a simple Level III model.	58

1 Introduction

In recent years, the deployment of electric vehicle (EV) charging stations has increased dramatically. In October of 2011 there were approximately 2,000 EV charging stations installed in the United States and Canada. Today, there are 24,034 public charging stations connected to the electric power grid in the United States alone.[23, 28] With the continued release of new consumer EVs to the market, this trend only shows signs of increasing.

EV chargers are a new and atypical load that is being presented on power distribution feeders. These non-linear loads often introduce power quality (PQ) issues within distribution circuits, which can have detrimental effects on system components. PQ is a measure of the fitness of electrical power from the utility to the electrical customer. PQ is of concern because it manifests as deviations in voltage magnitude, issues with continuity of service from the utility, or transient voltages and currents.[17] PQ also encompasses harmonic distortion, DC offset, phase imbalance, and voltage deviations. Of special interest are harmonic currents since these have the potential to affect the lifetime of magnetic assets such as distribution transformers and instrument transformers.

In order for distribution engineers to reliably predict the impacts that EV chargers have on distribution assets, they need access to models that represent the time-varying, unbalanced, and distorted nature of these loads. The purpose of our research is to develop a tool that allows distribution engineers to observe these impacts within a larger system

model, one that contains non-linear, time-varying elements itself. Such a tool can be used in the early planning phases of projects to aid in the design of distribution feeders; to assist in long-term asset planning by helping to predict asset lifetime; and to upgrade existing feeders in anticipation of changes caused by these EV load profiles. By modeling these time-varying, unbalanced, high-harmonic loads, designers will be better positioned to properly select assets designed to accommodate such loads over the long term.

The objectives of this work are twofold, one: to expand the electric utility industry's understanding of the issues that electric vehicles cause to distribution systems, thereby enabling them to protect costly assets, and two: to construct a tool that provides distribution engineers with a platform on which to create system designs that take into consideration the adverse effects of EV chargers.

1.1 Background

Utilities plan asset management by anticipating the nature of loads and selecting assets designed to handle those loads. A deeper understanding of these matters specific to EVs will aid utilities in the design of distribution systems and provide guidance for asset planning. A load's PQ particularly affects magnetic assets because of the potential for insulation failure and core saturation. Understanding the PQ of non-linear loads assists distribution engineers with the selection of k-factor¹ ratings for distribution transformers, selection

¹Specified in ANSI/IEEE C57.110, k-factor denotes a transformer's ability to serve non-linear loads without exceeding temperature limits.

of current transformers (CTs) and voltage transformers (VTs), protection settings, and decisions regarding conductor ampacity.[2]

The impacts that harmonic distortion can have on distribution assets, particularly transformers, power cables, capacitors, metering, relaying, and switch gear can be detrimental to the proper operation of power systems. Harmonic distortion can also affect near-by loads, particularly power electronics devices and motors.

Harmonic distortion is a deviation of a current or voltage waveform from a perfect sinusoidal profile. In the case of non-linear loads, such as EV charge controllers, current distortion is very common due to the use of power electronics switches to convert power from an AC to a DC form. Introduction of these distorted currents into the distribution system can distort the utility supply voltage and damage electrical distribution equipment, particularly magnetic components. In order to help mitigate these adverse affects, the IEEE established Standard 519-1992, with the objective of developing “recommended practices and requirements for harmonic control in electrical power systems”. [1] This standard describes the problems that unmitigated harmonic current distortion can cause within electrical systems as well as the degree to which harmonics can be tolerated by a given system. The standard recognizes the responsibility of an electrical user to not degrade the voltage of the utility by drawing heavy, distorted currents.[15]

1.1.1 Transformers

Current harmonics can be especially troublesome for power transformers. One example of the losses caused by high harmonic content in the system is I^2R losses. These losses

are due to higher-order currents within the transformer windings. If the root mean square value of the load current is increased due to a harmonic component, the I^2R losses increase accordingly.[25] Consequently, the transformer will consume more real power than anticipated, reducing its efficiency in conveying power to customers.

Another concern in the presence of increased harmonics are eddy current losses and hysteresis in the core of the transformer. These issues create abnormal temperature rise in the transformer. This increased temperature accelerates the degradation of insulation within the transformer, ultimately leading to a shortened life span for the equipment.[10] Eddy current and hysteresis losses are frequency dependent, so higher order harmonics are particularly problematic for transformers.

Other losses due to increased harmonic content are stray flux losses. These can occur in the core, clamps, tank and other iron components of the transformer. Stray flux induce eddy current and hysteresis losses in these components, leading to increases in the oil temperature and thus hot spot temperatures within the transformer. This can also contribute to the premature degradation of the transformer insulation and oil, leading to eventual catastrophic failure of the equipment.

1.1.2 Power Cables

The primary effect of harmonics on power cables is the additional heating due to an increase in the I^2R losses. The increase in power dissipation across a conductor can be attributed to the two phenomena known as the proximity effect, which results in current crowding, and the skin effect, which forces the electric current to flow on the outer surfaces of the line. Both of

these cause an effective increase in the resistance of the line that is proportional to the higher frequencies of the harmonic components. Also, cables involved in system resonance may be subjected to voltage stress and corona, which can lead to dielectric (insulation) failure.[29]

1.1.3 Relays, Switch Gear, and Metering Equipment

Protective relaying equipment, switch gear, and metering equipment may also be negatively impacted by the presence of harmonic currents. Relaying equipment may operate more slowly because of higher pick-up values than settings would otherwise dictate, resulting in unexpected operation. Fuses may experience premature operation due to I^2R heating by harmonics. And as with power transformers, harmonic currents can increase heating in CTs and VTs due to I^2R , eddy currents and core saturation, leading to shortened asset lifetimes. Within switchgear, the presence of harmonics contributes to I^2R heating, reduces steady-state ampacity, and shortens lifetimes of insulating components.

1.1.4 Capacitors

Harmonics introduced by a non-linear load may interact with nearby capacitors if the harmonic frequency is in resonance with a LC time constant. The inherent positive reactance of distribution cabling, transformers and loads, can couple with the negative reactance of capacitor banks, resulting in very high voltages and currents at resonant frequencies. The unexpected increased voltage stress and I^2R heating within resonating capacitors can result in a shortened asset lifetime or catastrophic failure.

1.1.5 Total Harmonic Distortion

EVs employ power electronics within the charge controllers that interface the vehicle's electric power system with the grid. For Level 1 and Level 2 chargers, vehicle charging is done by an on-board AC-DC controlled rectifier that couples with the electric service via a single phase connector. For Level 3 chargers, aka DC Fast Chargers, the charging is controlled by electronics within the charge controller.[24] In either case, the harmonic distortion introduced into the distribution system by these charge controllers can be measured in terms of Total Harmonic Distortion, THD. However, it should be noted that the THD of a charger changes throughout the charging cycle as the firing angles of the power electronics switches changes in response to the various phases of the charging cycle. Further, the THD on a utility feeder is compounded when multiple EVs are connected to the same service. Equation 1.1 illustrates how the THD for each charger is calculated.

$$I_{THD} = \frac{\sqrt{\sum_{n=2}^{\infty} I_n^2}}{I_1} * 100\% \quad (1.1)$$

1.1.5.1 Total Demand Distortion

While the IEEE 519-1992 standard recommends limits for total current harmonic distortion within a distribution system, the metric Total Demand Distortion (TDD) is based on the size of the load with respect to the size of the power system to which it is connected. The maximum allowable TDD is determined by the ratio of the short circuit current at the point of common coupling (PCC) to the average maximum demand load current for the system for

the previous 12 months (I_L).[15] This dictates the percentage of harmonic current that the user is allowed to inject into the utility system. As the size of the load increases with respect to the size of the system, a lower percentage of harmonic current injection is allowed. This is a challenge for EV charging loads since they can draw extremely large currents, and the TDD of such loads changes during the course of a charging event. Ideally, the harmonic distortion caused by a single consumer should be limited to an acceptable level at any point in the system. However, the prescribed levels for TDD establish the maximum allowable current distortion for a given system.[1] Equation 1.2 shows how the TDD for a system is calculated.

$$I_{TDD} = \frac{\sqrt{\sum_{n=2}^{\infty} I_n^2}}{I_L} * 100\% \quad (1.2)$$

1.1.6 Triplen Currents

Non-linear loads can induce triplen currents within three-phase systems. Triplen currents are comprised of the non-even multiples of $3n$, known as the triplen harmonics.[7] Examples of these are the 3^{rd} , 9^{th} and 15^{th} harmonics. Triplen currents are troublesome because they add up in the neutral line of a grounded wye configured system. Similarly, triplen currents circulate in the case of a delta wired system. When these triplen currents superpose in the neutral line, they can cause excessive currents and can lead to conductor heating.[14]

1.1.7 System Imbalance

When system imbalance occurs the current and voltage in one phase differs from that in another. This can cause what are known as zero and negative sequence components, which are a measure of the imbalance in polyphase circuits. These currents can cause loading on the phases and neutral line that is greater than they are designed for. Often, in installations with single-phase, two-wire chargers there are problems with system imbalance when various chargers are coming on and off line at different times. When one phase is heavily loaded, another will be loaded more lightly. In order to mitigate these problems they must first be quantified. The IEEE 1459-2010 standard defines the method for calculating electric power under unbalanced conditions.[3] When the traditional vector apparent power calculation is replaced with the effective apparent power definition presented by the standard, the system can be quantified in a manner where power factor is correctly computed for unbalanced loads.

1.1.7.1 Power Resolution Tree

The current harmonics in three-phase non-sinusoidal situations can be evaluated using the IEEE Standard 1459-2010.[3] This standard quantifies the active and reactive powers in a three-phase unbalanced system, as shown in Figure 1.1, based on the effective apparent power for the system, S_e . The standard goes further to break down the definition of power into fundamental and non-fundamental components (S_{e1} and S_{eN} , respectively), positive sequence components (S_1^+ , P_1^+ and Q_1^+) and system unbalance as quantified by fundamental

unbalance power S_{1u} . Finally, harmonic active (S_{eH} , P_{eH} and D_{eH}) and distortion (D_{eI} and D_{eV}) powers are described as well.

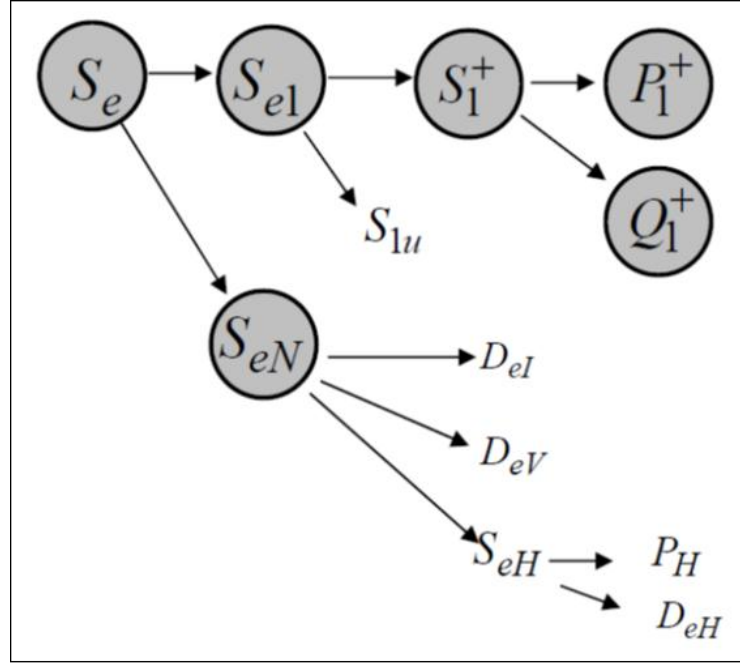


Figure 1.1: The power resolution tree for three-phase non-sinusoidal conditions.[11]

1.2 EV Charger Models

It is well known that the load behavior of EV charging units is problematic for power systems and the components associated with them.[9, 26, 31] And, while there have been various attempts to model these chargers, each charger design is different from the next and usually contains some sort of proprietary design, making the modeling task more than trivial.[8, 13, 21] There are certain aspects of the characterization of these chargers that has proven difficult to present in a generalized fashion that is useful across different system designs.[6, 13, 16, 22] Also, there are currently no freely accessible tools or power system design software add-ons available to power and distribution system designers.

The solution proposed here treats the EV charger models not as constant power elements, but considers their state of charge (SOC), which changes throughout the charging cycle, in the design. These charger models are not created by using a generalized circuit, as is popular in the current body of work, but uses actual harmonic load data in order to construct analytical solutions that behave as non-linear impedances.[6, 8, 9, 13, 16] This makes the load models responsive to the applied sinusoidal voltage of the system which, in the behavioral model, illustrates the effects, such as current, THD, and power flow, that the chargers have on distribution systems and their assets of interest. The discrete set of SOC's allows the user to analyze the system at various stages in the charging cycle, while forgoing the extremely computationally intensive process that would be required of a fully dynamic model. This saves processing resources and, more importantly, time.

As part of this work power system design tool has been created that is available on the

systemvision.com website. It is a free and collaborative environment where distribution system designers can layout, and analyze the response to, any system of interest, especially ones that include power system, three-phase, and EV charger components. The flexibility of these models also enables the user to examine existing, proposed, and retrofitted distribution systems that may include EV chargers. There is the important ability to examine chargers in aggregate as well as run different simulations for chargers coming on and off line at various points in their charging cycles. Based on the user's needs, specific aspects of power quality can be examined for a given system such as power flow, current content, THD, TDD, and load imbalance effects. This enables the user to not only utilize the tool to inform them in their design decisions, it can assist the designer in calculating values for transformers, conductors, and protection equipment.

2 Data Collection

Electric Avenue, located on the Portland State University (PSU) campus along SW Montgomery St. between 6th Avenue and Broadway, is a joint project between Portland General Electric, PSU, and the City of Portland.² Launched in August 2011, Electric Avenue is intended as a research platform for understanding the impact electric vehicles have within the larger context of the city. For this research, Electric Avenue was utilized in order to gather PQ measurements from the electric vehicle chargers, paying specific attention to the THD of individual EV chargers and the TDD of the Electric Avenue service. The phase imbalance, parasitic loading, and other PQ issues observed during the course of the study were also noted.

2.1 Measurement Procedure

At the time of data collection, the site had five Level 2 chargers and two Level 3 chargers, which were donated by six different manufacturers.³ The Level 2 units are alternating current (AC), single phase, two-wire machines that, when attached to an EV manufactured with a SAE J1772 charging receptacle, replenish the EV's battery with a 4 to 20 kW input at 208 V.[5] Depending on the vehicle type, it can take up to 8 hours or more to fully replenish

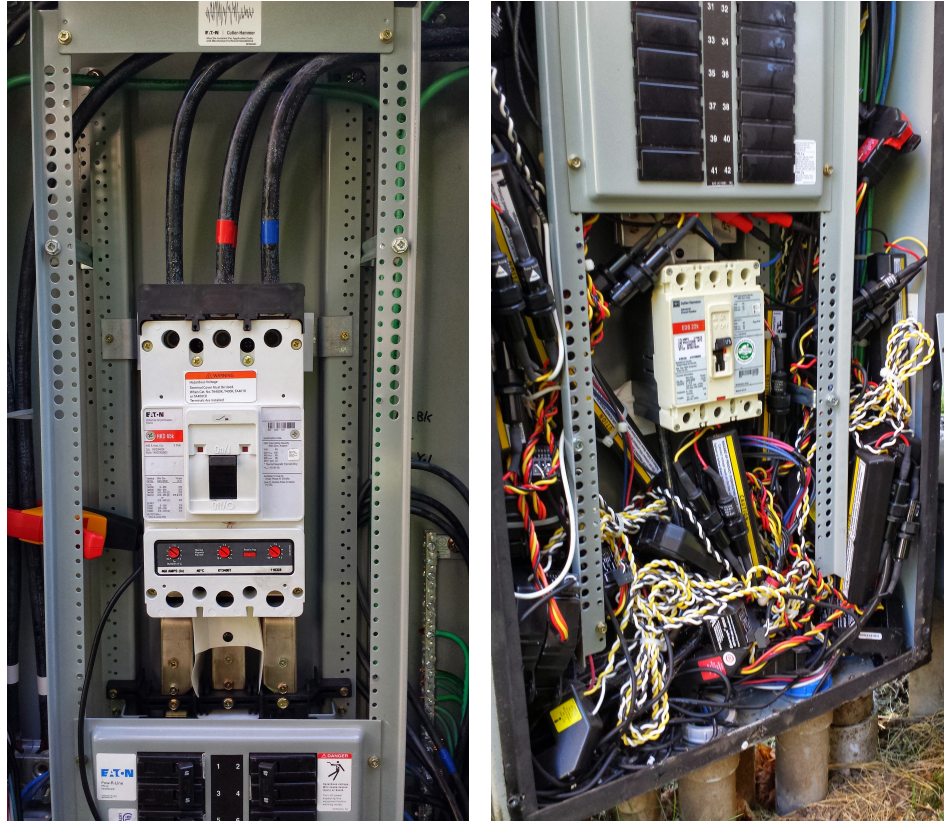
²Electric Avenue website: www.pdx.edu/electricavenue

³EATON, GE, Kanematsu, OpConnect, Shorepower, and SPX.

a depleted set of batteries.[30] The power electronics that control the flow for the Level 2 type chargers are located on board the vehicles themselves. In contrast, the Level 3 charging units are three-phase designs that deliver power through a CHAdeMO standard receptacle. These are capable of delivering between 20 and 50 kW via direct current (DC) and can recharge a set of EV batteries in as little as 30 minutes.[30] The power electronics that control the power for the Level 3 chargers are located off-board the vehicle, within the charging unit itself.

During the day of November 14th, 2013 a Tektronix PA4000 three-phase power analyzer was connected at the site. CTs and voltage clamps were connected to the branch circuits of each of three chargers, including the SPX, ShorePower, and OpConnect branded units, within the service panel. Data were recorded during real-time charging events. The PA4000 produced one data set every half second, which consisted of the magnitude and phase angle of both the current and voltage for each of the first through twenty-first odd harmonics. The site was monitored during data collection, so charging events were correlated with EV model types for each event that occurred. Figures 2.1(a) and 2.1(b) show both the existing service entrance and the metering equipment that was temporarily installed at the site for data collection.

The data collected were exported from the PA4000 utilizing the Tektronix PWRVIEW Analysis software to Excel .xlsx files via a laptop at the site. The data sets grew large enough to exceed the processing capabilities of the laptop in less time than a full vehicle charging cycle. To compensate for this, sets of data were exported approximately every 30



(a) Service entrance at Electric Avenue. (b) Existing metering equipment at Electric Avenue.

Figure 2.1: The service entrance at Electric Avenue with a red and yellow CT installed for data collection can be seen on the left. On the right, CTs and PTs permanently installed at the site.

minutes and later stitched together, using a desktop system, into larger .xlsx files. Since information stored in this format is highly flexible and, therefore, can be utilized by many different types of software, it proved to be most valuable for utilization in analysis and modeling applications.

3 Design Methodology

Several power system simulation environments were considered for this work, but none offered the ability to model these time-varying, harmonic loads. Rather, a tool was used that uses the versatile and capable IEEE Standard hardware modeling format, VHDL-AMS. IEEE Standard 1076.1 (VHDL-AMS) is a super-set of the IEEE Standard 1076 (VHDL), a widely used Hardware Description Language for digital systems. In addition to digital behavior, VHDL-AMS provides modeling capability for analog and mixed-signal hardware. SystemVision, the modeling, simulation, and design analysis platform from Mentor Graphics, provides the ability to create custom, highly configurable models of (analog) power distribution system elements, such as transformers, transmission lines, circuit breakers, and induction motors in addition to the abstract harmonic load models of the EV chargers.[4]

3.1 Models and Schematics

A description of the particular design parameters for each component used in the system designs is included below. Examples of the VHDL-AMS code for each design can be found in Appendix B.16.

3.1.1 EV Charger Components

In order to accurately represent an EV charger, a model must present the impedance that changes as a function of time relative to where the EV is in its charging cycle in terms of both its power draw and its harmonic profile. As the battery gets closer to a full charge, an EV charger adjusts the firing angle of the power electronics within the charge controller. This causes the controller to decrease the power being delivered in order to not overcharge the battery, thereby extending EV battery life. As the power electronics adjust in order to start the trickle-charge state, the magnitude of the harmonic contributors increase in relationship to the fundamental component. It is these periods of high harmonic loading that are of most interest due to their effect on distribution system assets. In order to represent these loads in a model, VI data from discrete times throughout the charging cycle were selected. How a single charger presents itself on a three-phase power system at various points throughout the charging cycle can then be analyzed. Further, the response of the system to multiple chargers coming off and on line, and at various impedance states can then be observed. These models were created as active rather than passive components in order to accurately simulate their non-linear impedances in response to the applied voltage on the system.

In order to simulate the chargers at representative points throughout their cycles, sets of current magnitude and phase angle readings through the fifth harmonic were utilized. The even harmonics were ignored as they are only found to be contributors to the waveform when there is x-axis asymmetry present. This asymmetry across the x-axis is not present

in EV chargers that are functioning properly. The current magnitudes and phase angles were selected from representative ranges of time throughout the charging cycle and are illustrative of the changing levels of THD presented to the system by the charger. The set of distortion levels selected for the Level 2 charger model included THDs of 3%, 4%, 8%, 10% and 12.5%. In order to greatly simplify the mathematical calculation required to process the data, the source voltage is assumed to be a root mean square (RMS) value of 120 volts. While there are some harmonic contributions to the fundamental voltage, they only have total contributions of less than 2%, making them insignificant enough to ignore. This is also based on the assumption that the utility provides a constant voltage that is within standard operating tolerances.

3.1.1.1 Analytical Solution

In order to utilize the current harmonic data to create a load that behaves as non-linear impedance, as opposed to a current source, a modeling method was developed to make the load responsive to the applied sinusoidal voltage. The voltage across the terminals of the load is read, time-domain mathematical processing of the voltage is performed, and the result is the value of the current that flows through the device terminals.

A graphical representation of a simplified version of the AC harmonic load model is shown in Figure 3.1. An input sinusoidal time-domain voltage, $v(t)$, is applied across the model's terminals, from p1 to p2. The voltage is scaled by $1/V_p$, the inverse of the amplitude (or peak voltage) of the sinusoidal stimulus at which the original load was calibrated. Then this "normalized" time-domain voltage $V_n(t)$, with peak value of approximately 1.0, is

integrated, differentiated, and scaled by gain coefficients k_{1i} , k_{1d} , and k_1 . Similarly, the voltage is raised to the third power and the result is again integrated, differentiated, and scaled by gain coefficients k_{3i} , k_{3d} , and k_3 . These results are summed and the value is the current that is then forced to flow between the model's terminals, from p1 to p2.

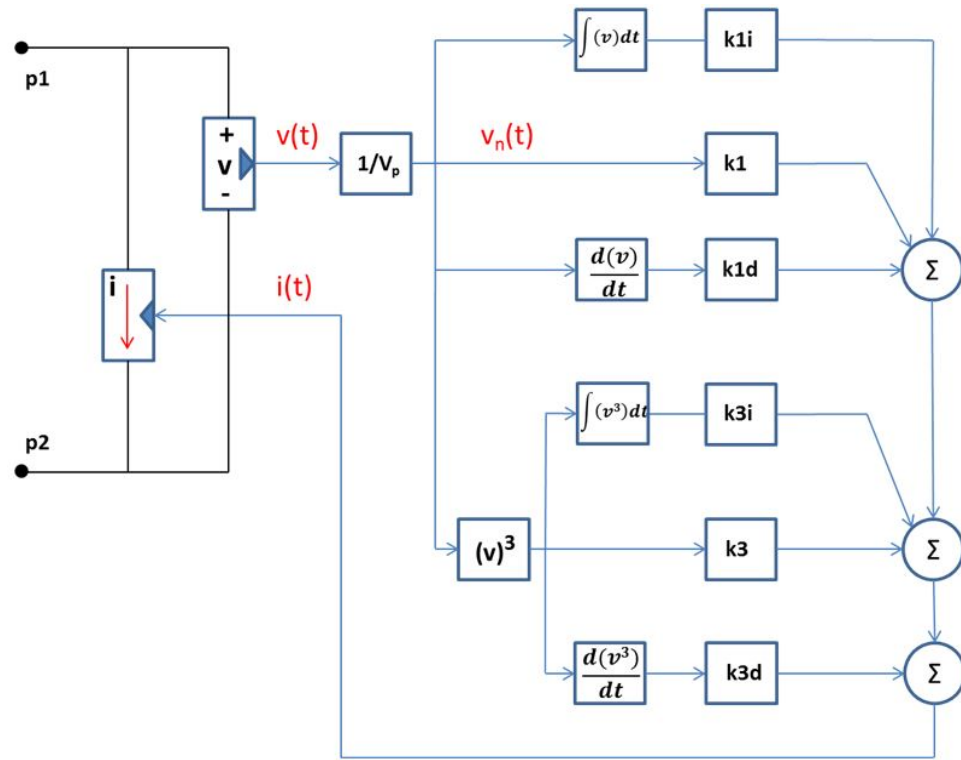


Figure 3.1: A graphical representation of a simplified version of the AC harmonic load model.

The corresponding analysis of the model's time-domain current response to an applied sinusoidal voltage at frequency ω (radians/sec), is shown in Figure 3.2. As can be seen from the analysis, the cubing operation gives a non-linear characteristic that results in third harmonic components in the observed terminal current. The six gain coefficients can be selected to fit the desired user specified spectral content, for both the fundamental and third harmonic in this case. Only 4 coefficients are actually needed to match the desired

magnitude and phase of these two harmonics. But having 6 allows the selection algorithm to assign two of the set {k1i, k1d, k3i k3d} to zero, so the remaining non-zero gains are positive. This is important for the reactive current contributors, for simulation stability.

```

In[4]:= v[t_] := Cos[ω * t];
i[t_] := k1i * Integrate[v[t], t] + k1 * v[t] +
k1d * D[v[t], t] + k3i * Integrate[v[t]^3, t] +
k3 * v[t]^3 + k3d * D[v[t]^3, t];
Expand[TrigReduce[i[t]]] // TraditionalForm

Out[4]//TraditionalForm=

```

$$\begin{aligned}
& k1 \cos(t \omega) - k1d \omega \sin(t \omega) + \frac{k1i \sin(t \omega)}{\omega} + \\
& \frac{3}{4} k3 \cos(t \omega) + \frac{1}{4} k3 \cos(3 t \omega) - \frac{3}{4} k3d \omega \sin(t \omega) - \\
& \frac{3}{4} k3d \omega \sin(3 t \omega) + \frac{3 k3i \sin(t \omega)}{4 \omega} + \frac{k3i \sin(3 t \omega)}{12 \omega}
\end{aligned}$$

Figure 3.2: Analysis of the model's time-domain current response to an applied sinusoidal voltage at frequency ω , using Mathematica.

The applied voltage amplitude and frequency must be approximately equal to the stimulus voltage at which the current harmonics were calibrated. For that stimulus, this behavioral model accurately represents the complex, non-linear load from which the current harmonic data was obtained, but it cannot predict the current response under other stimulus conditions. It does, however, provide the correct load current regardless of the phase of the voltage stimulus, as the time-domain current is dependent on the applied voltage; it is not an independent current source or sink.

3.1.1.2 Behavioral Solution

The behavioral solution is implemented within the SystemVision software environment. Two-terminal symbols are used to represent the ports across which the stimulus voltage is applied and the current is forced to flow. Each model contains multiple sets of harmonic load data that correspond to different charging conditions or “load states” for the given EV charger it represents. The user can select the particular operating state for each individual instance of the model that is included in the power system schematic, for any single simulation. Multiple simulation runs can be made to cover each loading configuration to be analyzed for the power distribution network. They will provide a time-domain window of the currents, THD, and power-flow that will occur under each of those loading conditions, and the effects that the chargers have on the system, and its assets, can be analyzed.

3.1.2 Three-Phase Test System

The following is a description of the components that were utilized to create the three-phase test distribution feeder system. The construction of the systems is then described and an example of such a system is shown.

3.1.2.1 Infinite Three-Phase Source

The three-phase voltage drive is Y-connected with an exposed neutral terminal that is connected to ground for the purpose of our simulations. The model can be configured as an imbalanced three-phase source, for systems that may require this, by defining different amplitude and resistance values for each phase. The effective power output is then internally

calculated with these values. The default value for this component is $277 V_{RMS}$. The starting phase for phase A can also be manually set; the other phases are then automatically adjusted by the same amount, always 120° out of phase from one another.

3.1.2.2 Three-Phase Transmission Line

The three-phase transmission line model incorporates resistance with self and mutual inductance effects. These values can be manually adjusted by the user. Offset resistance and self & mutual inductances can also be defined based on the transmission line configuration to be examined. The length of the line is input in the units of meters. With these values the model internally calculates the effective power transferred and dissipated by each phase and in total.

3.1.2.3 Delta-Wye Transformer

The delta-wye configured transformer model behaves as a perfect transformer at all frequencies. It can be configured as an imbalanced transformer by setting different resistance values for each of the legs on the winding sets. The number of turns for each side can be set and the direction of power flow through the transformer is set by changing the sign for the associated generic. The model internally calculates the effective power transferred and dissipated using these values.

3.1.2.4 Delta-Delta Transformer

The delta-delta configured transformer model, similar to the delta-wye transformer design, behaves as a perfect transformer at all frequencies. It can be configured as an imbalanced transformer by setting different resistance values for each of the legs on the winding sets. The number of turns for each side should be set to define the voltage ratio. The model internally calculates the effective power transferred and dissipated using these values.

3.1.2.5 Single-Phase Center-Tapped Transformer

The single-phase transformer model behaves as a perfect transformer at all frequencies, including DC. It can be configured as an imbalanced transformer, by defining the turns for each secondary output, so that they each supply different power outputs. The model has an exposed neutral that should be tied to ground. The model internally calculates the effective power transferred and dissipated using these values.

3.1.2.6 Cable

The cable model uses wire resistance data from the American wire gauge (AWG) table. The user selects an AWG standard integer value for the wire size to be used in the system (a value from 0 to 40) as well as entering the length, in meters, of the wire. The model uses these values to internally calculate the voltage drop across it.

3.1.2.7 Schematic Design

An example of a radial distribution test system with multiple chargers can be seen in Figure 3.3. The three-phase infinite source, transmission line and delta-wye transformers comprise the distribution portion of the system. There are cables and single-phase transformers delivering power to the connected EV charger loads. In this case there are two Level 2 EV charger loads connected to each phase of the distribution lines.

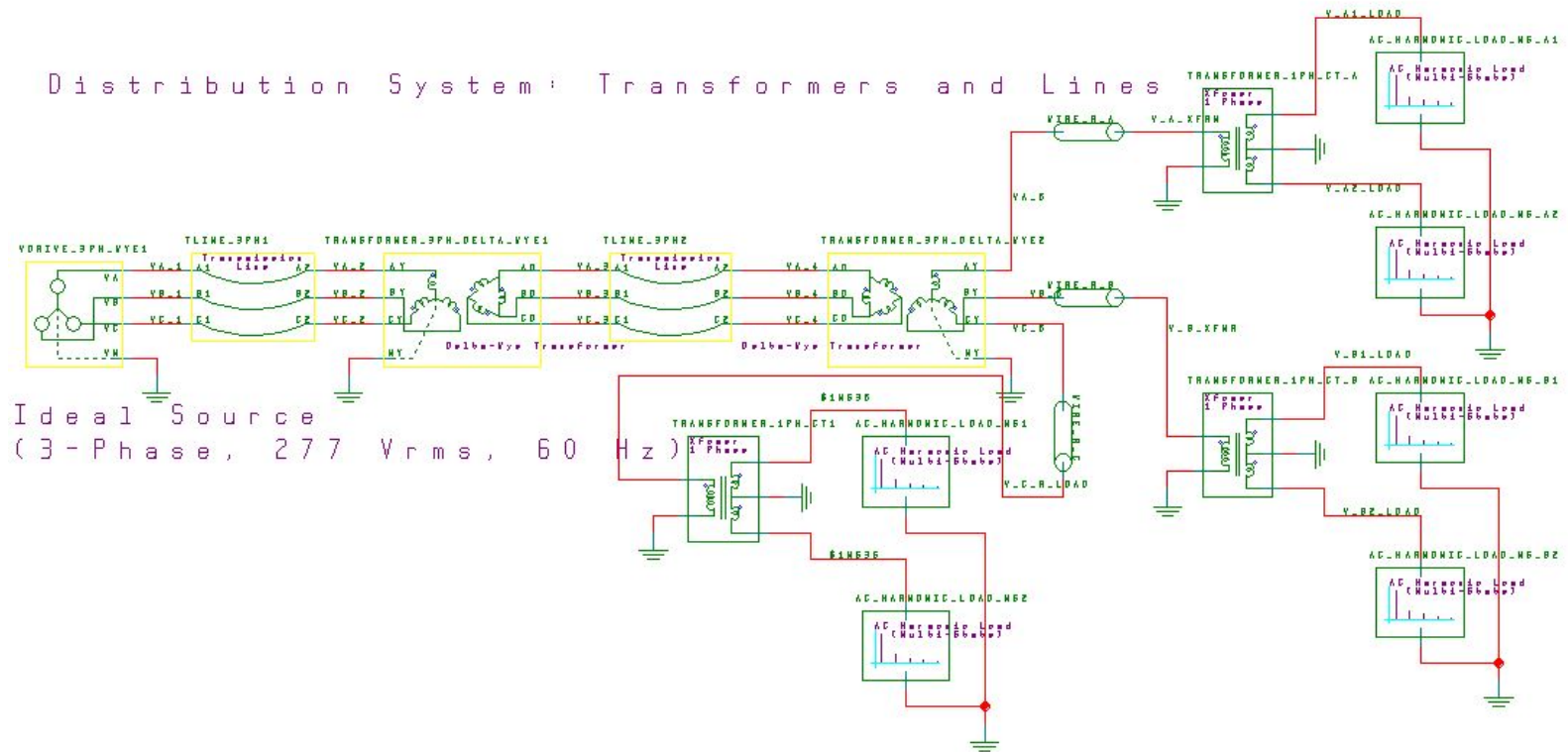


Figure 3.3: This is an example of a radial distribution test system with multiple Level 2 chargers attached. This schematic is used to run simulations and analyze the power flow throughout.

3.1.3 IEEE 13 Node Test System

Based on the complete data for a four wire wye-connected distribution test feeder published in a 1992 paper, the IEEE PES Distribution System Analysis Subcommittee's Distribution Test Feeder Working Group then published another paper in 2001 that was an updated version of the original paper.[19, 20] The purpose of the updated paper was to provide a set of data that could be used by distribution engineers to validate their designs by being able to verify the correctness of their model. All of the papers and spreadsheets with system values are maintained and available for download on the committee's site.[27] For the case of validating the EV charger designs, this proves to be immensely valuable. Another group utilized the test system to validate their VHDL-AMS environment as part of the initial development for a real-time digital simulator for electrical power distribution feeders using structures to describe the hardware and the software through FPGA devices.[18] While their second paper is still pending, the initial paper illustrates the successful use of the 13 node system as a verification tool for the VHDL-AMS language.

3.1.3.1 Voltage Source

The voltage source component is a standard component from the VHDL-AMS electrical library. For the purpose of the IEEE 13 node system, the frequency was set to 60 Hz and the voltage was set to 66.3953 kV. This produces a voltage of 115 kV from line to line. The sources are each out of phase by 120°.

3.1.3.2 Single-Phase Transformer

For the substation transformer model the standard component from the VHDL-AMS electrical library was used once again. This is an ideal transformer where perfect coupling with a coefficient k of 1 is assumed. In this case the primary and secondary winding resistances and inductances were calculated from the values for the primary and secondary voltages of 115 kV and 4160 kV, respectively, the system frequency of 60 Hz, and the resistive and reactive values of 1% and 8%, respectively. The per unit resistance values were multiplied by the base impedances for each side of the transformer in order to obtain the full resistance values for either side. The per unit reactance values were also multiplied by their respective base impedances in order to get their full values, then converted into terms for inductance using the ω for the system. These calculations accurately portray the transformer model as described for the construction of the test system.

3.1.3.3 Voltage Regulator

The last component in the substation portion of the design is the voltage regulator, which is prescribed to hold the system voltage at 122 V. This component has an output voltage set point of the prescribed value, a minimum input voltage that is defined to be at least half of the output voltage set point, and a maximum output current just larger than the rated ampacity of the lines. While this design would never realistically work in application, it serves as an ideal model with which to simulate the required outputs.

3.1.3.4 Lines

The transmission lines within the system are modeled with simple series resistors and inductors from the VHDL-AMS electrical library. The calculations for each of the lines is based on the length of the line segment, the conductor size, and the line configuration as described in the radial distribution test feeders article. The values for resistance and inductance per mile are given in the body of the paper.

3.1.3.5 Shunt Capacitors

The shunt capacitors were modeled using ideal capacitors from the VHDL-AMS electrical library. The values calculated for them was, once again, based on the ω of the system and the given values of 200 VAR, in the case of the delta connected bank, and 100 VAR, in the case of the capacitor located in parallel with the constant current load.

3.1.3.6 Constant Power Loads

Constant power loads were required for the design in both delta and wye configured formations. These loads were designed using a spice model that calculates values for resistance and inductance based on the voltage seen across the terminals, thereby maintaining a constant power draw for each. Unique values were entered for the power and reactive power of each load prescribed for the system. Regardless of the voltage seen on the feeder, these elements will always draw the same amount of power.

3.1.3.7 Constant Current Loads

Constant current loads are also required for the design in delta and wye configured formations. A constant current source from the VHDL-AMS electrical library was installed and reversed in order to simulate these loads. The current setting for the device was derived from the 170 kW and 80 kVAr values given for the load and based on the expected 120 V for the system. Aluminum plants, and electrolysis operations are examples of these types of loads. These are simple models which are valuable for analyzing power flow through a system.

3.1.3.8 Constant Impedance Loads

Finally, the constant impedance loads, in both wye and delta formations, were represented by simple series resistors and inductors from the VHDL-AMS electrical library. The values for the components were derived by relating the powers and reactive powers to the voltage of the system. These loads, along with constant power loads, are representative of the types of loads generally seen on distribution feeders.

3.1.3.9 Schematic Design

The VHDL-AMS schematic used for the analysis and validation of the IEEE 13 node test system can be seen in Figure 3.4. The substation section, with the sources, transformers, and voltage regulators can be seen on the left. The various wye and delta connected loads can be seen throughout the system.

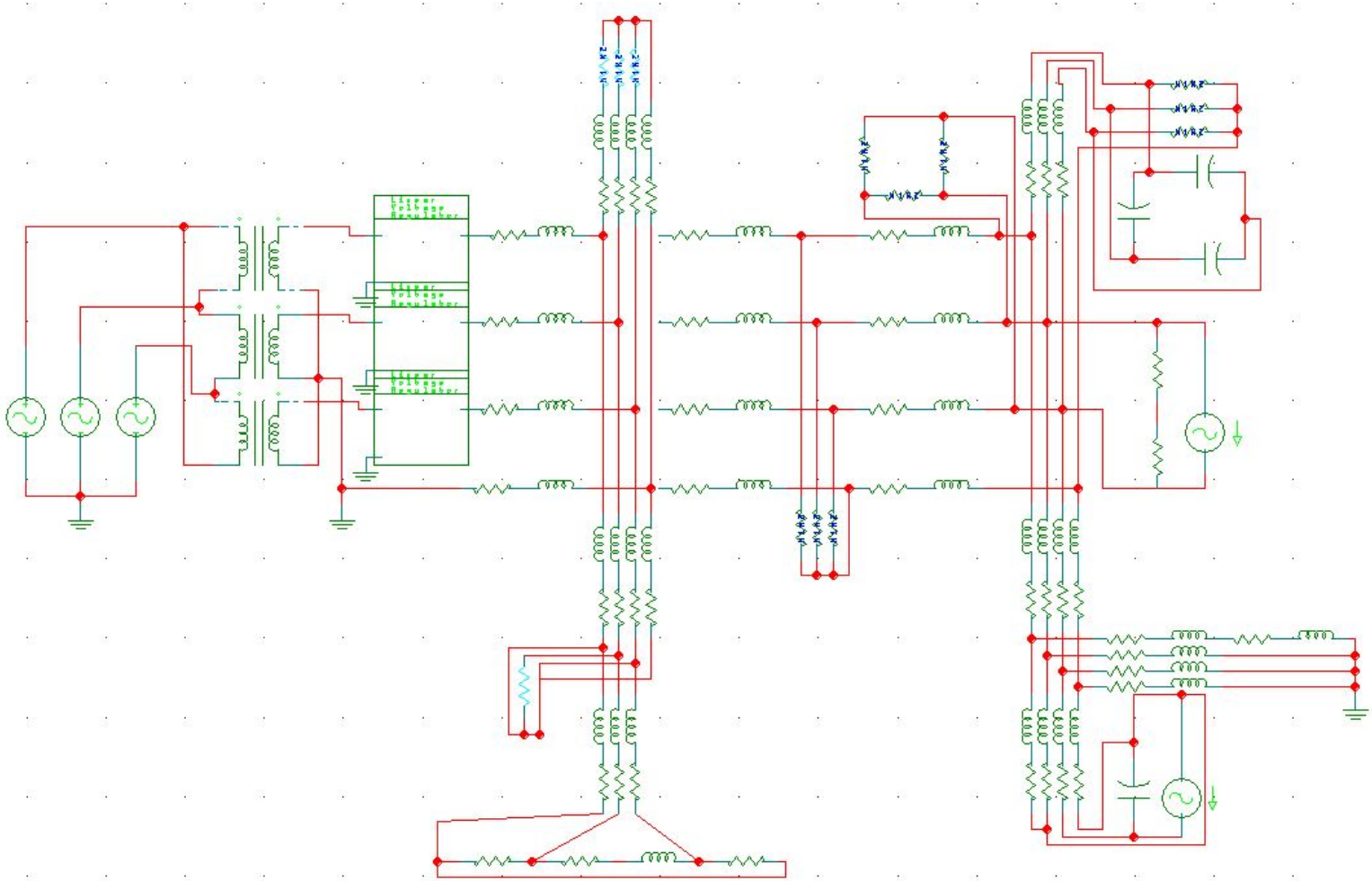


Figure 3.4: An example of the IEEE 13 node radial test distribution feeder as a schematic.

4 Results & Analysis

4.1 Simulations Analyzed

The following section presents a comparison of the raw data current signals in for the EV chargers as created in MATLAB versus the charger current signals, as modeled and simulated in the SystemVision environment, as well as Fast Fourier Transform (FFT) analyses of the charger models connected to a three-phase system in the simulation environment. The next section outlines the validation of the IEEE 13 node radial distribution test system within the SystemVision environment. Finally, an analysis of the behavior of the 13 node system, as the EV charger loads are added, is presented.

4.1.1 EV Chargers on Three-Phase Systems

For each of the charger models, Level 2 and Level 3, the raw data was first processed in MATLAB in order to show the original waveforms. Next, the VHDL-AMS charger models were simulated in SystemVision in order to draw comparisons between the two.

4.1.1.1 Level 2 Charger

Current waveforms as produced in MATLAB from the raw data for the Level 2 charger are shown in Figure 4.1. Each of the waveforms represents one of the five charging states.

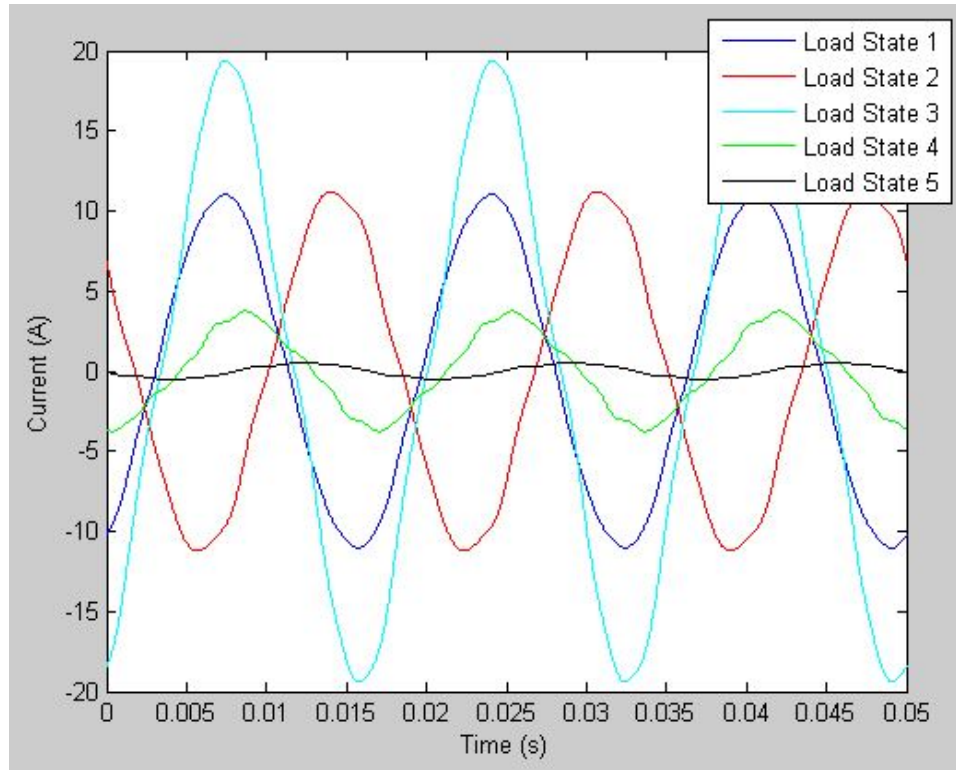


Figure 4.1: Overlay of currents of the Level 2 chargers at each of the five charging states created in MATLAB from raw data.

The current waveforms as produced by a simulation run for the Level 2 charger model in SystemVision can be seen in Figure 4.2. It should be noted that these waveforms are at least one second after the charger begins its simulation run. This is done to get past the transient start up period that occurs at the beginning of each of the simulation runs. The magnitude of the currents correlate with the current profiles produced in MATLAB. There are some slight variations in the waveform distortions that can be attributed to only the 3rd and 5th harmonics being included in the model while all of the harmonics recorded during data collection are included in the MATLAB waveforms.

Figure 4.3 shows the FFT results at the fundamental frequency for each of the five charging states of the Level 2 chargers overlaid on top of one another. The magnitude for

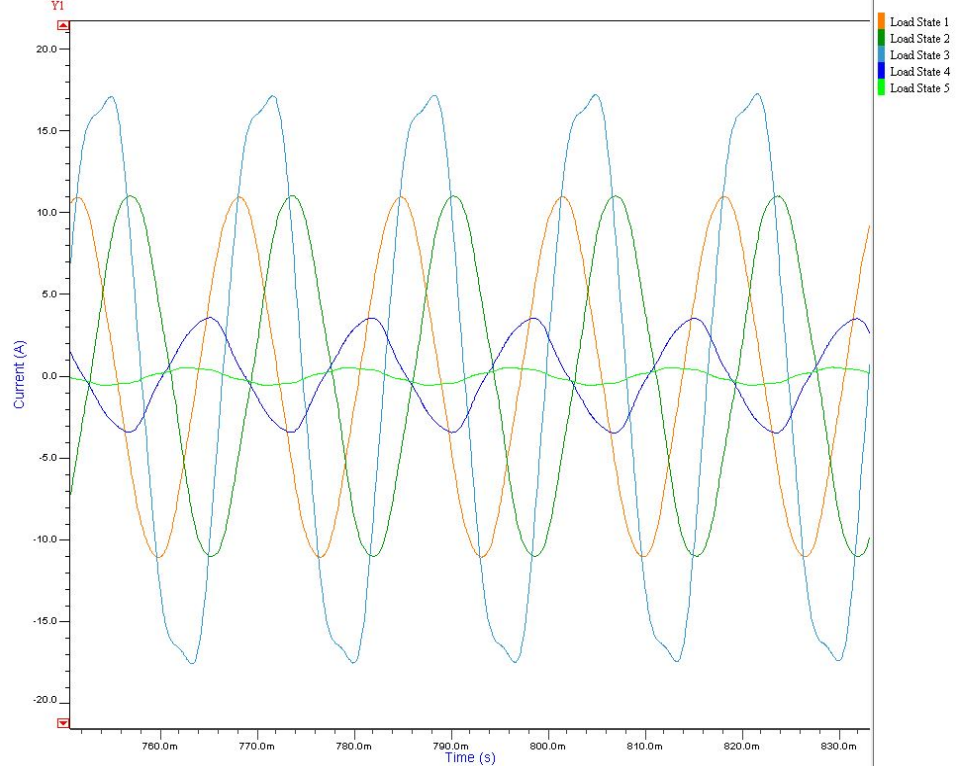


Figure 4.2: Overlay of currents of the Level 2 chargers at each of the five charging states simulated in SystemVision from VHDL-AMS model.

each charging state is indicated on the image. These magnitudes, and their associated phase values, correspond with the spectral content values for each charge state entered into the load model. While the phase values are entirely correct, the magnitudes are slightly less than anticipated. This is attributed to a short start-up transient that is included in the set of data processed by the FFT algorithm. When the the RMS current value of the time-domain waveform is measured, excluding the start-up transient, the peak value times $\sqrt{2}$ exactly matches the expected peak current. This indicates that the model accurately simulates the non-linear load profile based on user-defined data.

Figure 4.4 shows the FFT expanded out to include the 3^{rd} , and 5^{th} harmonics as well as the 1^{st} .

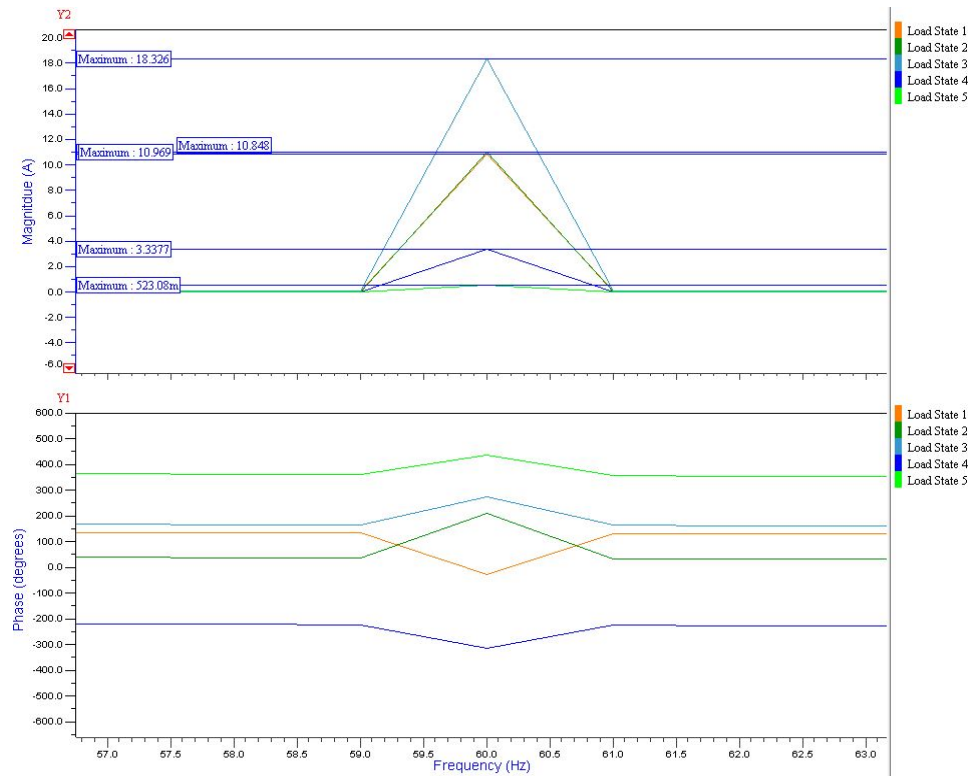


Figure 4.3: Overlay of FFT analysis of Level 2 chargers at the fundamental harmonic for each of the five charging states.

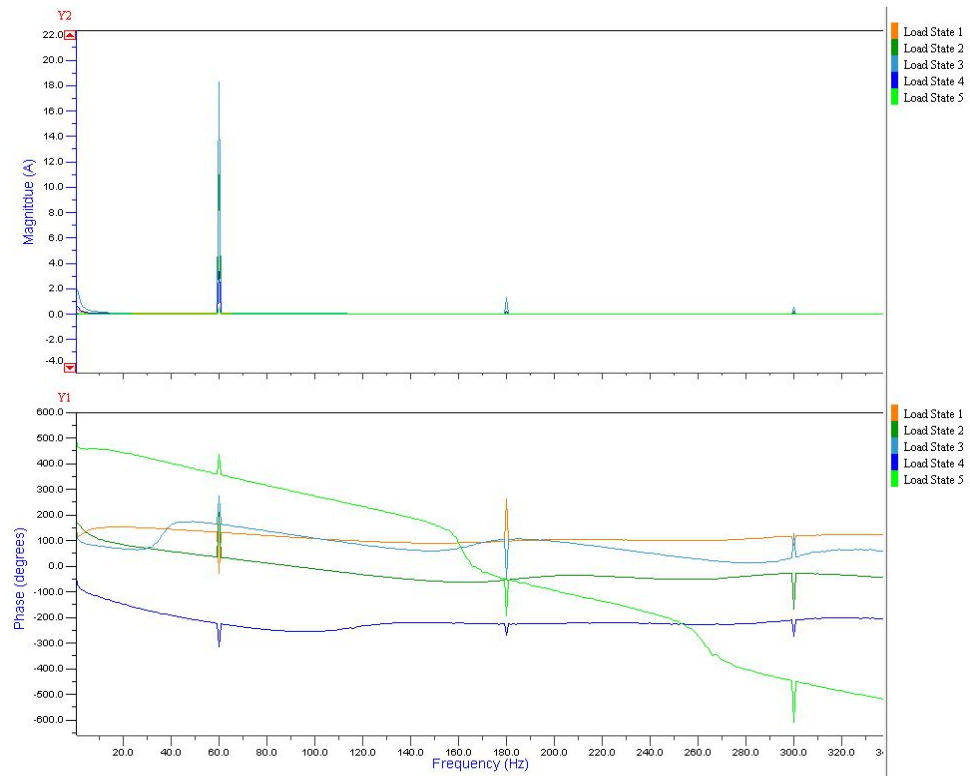


Figure 4.4: Overlay of FFT analysis of Level 2 chargers at 1st, 3rd, and 5th harmonics for each of the five charging states.

4.1.1.2 Level 3 Charger

The VHDL-AMS model design of the Level 3 charger is similar to that of the Level 2 charger. The layout of a schematic for a Level 3 charger can be seen in Appendix A.1. The same analytical and behavioral techniques as for the Level 2 charger are used in conjunction with the current magnitude and phase data collected for the Level 3 charger. Then, three of the models are linked together in a grounded wye configuration in order to simulate one charger. It should be noted that the state of charge for each of the three components that make up the model should be set to the same state or erroneous results will be produced. When connected to a three-phase distribution system, the currents for each of the components will be 120° out of phase with one another due to the source voltages.

Figure 4.5 illustrates the waveforms for each of the five charging states created in MATLAB from the raw data. Figure 4.6 shows the same set of waveforms as simulated from the Level 3 charger model in SystemVision. Again, the current magnitudes correspond. Also, similar to the Level 2 case, more distortion can be seen in the MATLAB output since all harmonics were included in that data set. In the case of the Level 3 charger, a reference angle of zero was used for each of the 1st harmonics for each charge state. The 3rd and 5th harmonics for each were then offset by their respective values. While this causes the entire waveform to be shifted along the x-axis it does not change the effect that the harmonics of the charger have on the system under simulation.

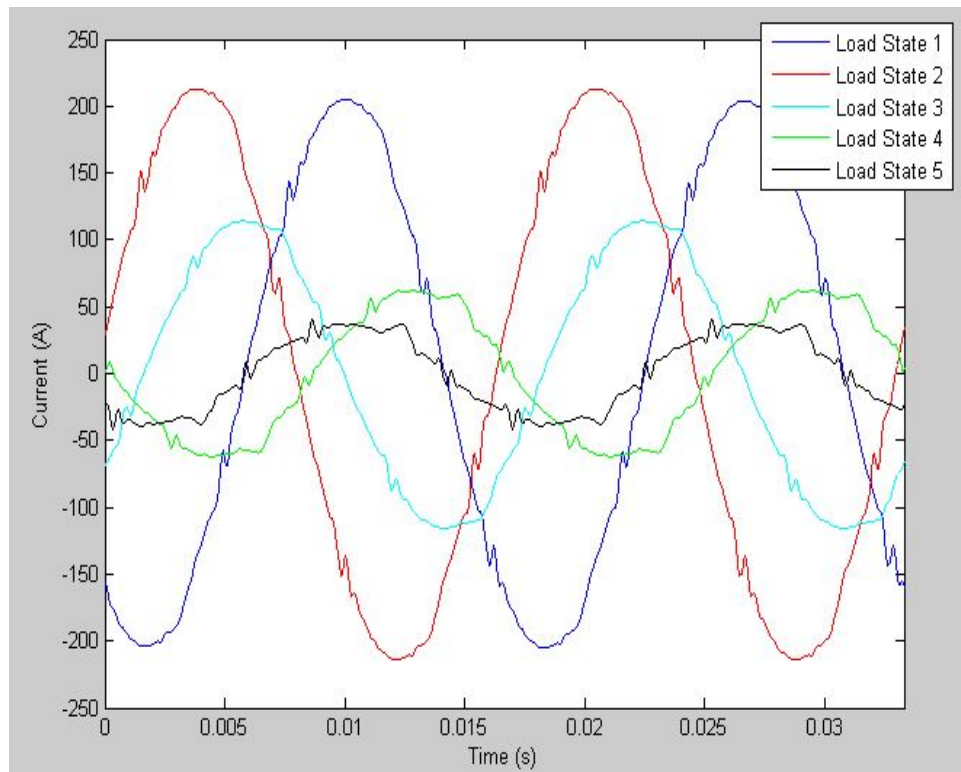


Figure 4.5: Overlay of currents of the Level 3 chargers at each of the five charging states created in MATLAB from raw data.

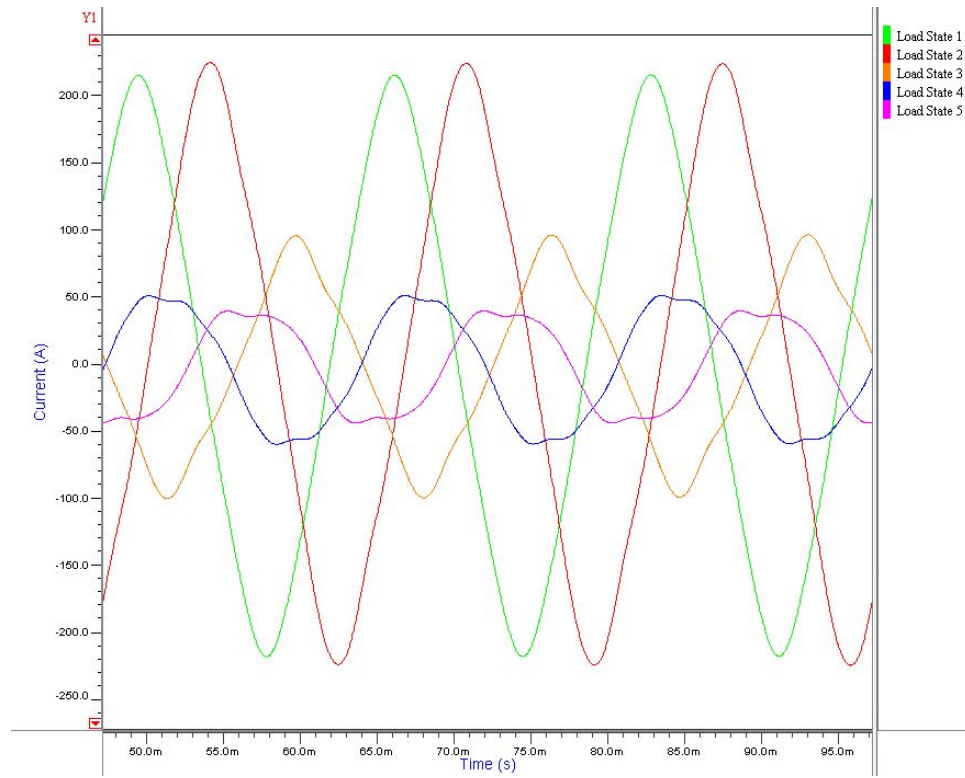


Figure 4.6: Overlay of currents of the Level 3 chargers at each of the five charging states simulated in SystemVision from VHDL-AMS model.

4.1.2 Validation of the 13 Node System

The 13 node radial distribution test system as shown in Figure 3.4 was utilized in order to validate the VHDL-AMS modeling environment. The loads on the system consist of three-phase balanced and unbalanced loads and single phase loads. The small and relatively highly-loaded system has a system voltage of 4.16 kV which is a very common distribution voltage. The IEEE 13 node system model is a standard model used for research. It typically presents no convergence problems.

The steady-state voltage profiles at each of the nodes was provided in the 13 node paper for comparison. Tables 4.1 and 4.2 show a comparison of the specified expected voltages and phase angles for each node in the system and the values from the VHDL-AMS simulation. Table 4.3 shows current between various nodes on the system. There are differences of less than 0.5% in all of the data sets, which is a strong validation of the VHDL-AMS model. The format for the comparison tables was inspired by tables that were created for similar validation of an ETAP 13 node system.[12]

Voltage (per unit)									
Bus	Phase A			Phase B			Phase C		
	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)
650	1	1	0.00%	1	1	0.00%	1	1	0.00%
632	1.021	1.021	0.00%	1.042	1.042	0.00%	1.0174	1.0174	0.00%
633	1.018	1.018	0.00%	1.0401	1.0401	0.00%	1.0148	1.0148	0.00%
634	0.994	0.994	0.00%	1.0218	1.0218	0.00%	0.996	0.996	0.00%
645	-	-	-	1.0329	1.0329	0.00%	1.0155	1.0155	0.00%
646	-	-	-	1.0311	1.0311	0.00%	1.0134	1.0134	0.00%
671	0.99	0.989	0.10%	1.0529	1.0518	0.11%	0.9778	0.9767	0.11%
680	0.99	0.989	0.10%	1.0529	1.0518	0.11%	0.9778	0.9767	0.11%
684	0.9881	0.987	0.11%	-	-	-	0.9758	0.9768	0.10%
611	-	-	-	-	-	-	0.9738	0.973	0.08%
652	0.9825	0.9815	0.10%	-	-	-	-	-	-
692	0.99	0.99	0.00%	1.0529	1.0529	0.00%	0.9777	0.9777	0.00%
675	0.9835	0.9835	0.00%	1.0553	1.0553	0.00%	0.9758	0.9758	0.00%

Table 4.1: Validation results for a comparison of the voltages given for the IEEE 13 node radial distribution test feeder and the voltages at the nodes in the VHDL system. The difference between the two is calculated as a percentage.

Phase Angle (degrees)									
Bus	Phase A			Phase B			Phase C		
	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)
650	0	0	0.00%	-120	-120	0.00%	120	120	0.00%
632	-2.49	-2.49	0.00%	-121.72	-121.72	0.00%	117.83	117.83	0.00%
633	-2.56	-2.56	0.00%	-121.77	-121.77	0.00%	117.82	117.82	0.00%
634	-3.23	-3.23	0.00%	-122.22	-122.22	0.00%	117.34	117.34	0.00%
645	-	-	-	-121.9	-121.9	0.00%	117.86	117.86	0.00%
646	-	-	-	-121.98	-121.98	0.00%	117.9	117.9	0.00%
671	-5.3	-5.25	0.04%	-122.34	-122.4	0.05%	116.02	116.1	0.07%
680	-5.3	-5.25	0.04%	-122.34	-122.4	0.05%	116.02	116.1	0.07%
684	-5.32	-5.31	0.01%	-	-	-	115.92	115.9	0.02%
611	-	-	-	-	-	-	115.78	115.8	0.02%
652	-5.25	-5.25	0.00%	-	-	-	-	-	-
692	-5.31	-5.31	0.00%	-122.34	-122.34	0.00%	116.02	116.02	0.00%
675	-5.56	-5.56	0.00%	-122.52	-122.52	0.00%	116.03	116.03	0.00%

Table 4.2: Validation results for a comparison of the phase angles given for the IEEE 13 node radial distribution test feeder and the angles at the nodes in the VHDL system. The difference between the two is calculated as a percentage.

Current (A)									
Bus	Phase A			Phase B			Phase C		
	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)	IEEE 13	VHDL	Diff (%)
611	-	-	-	-	-	-	71.2	71.2	0.00%
632 - 633	81.3	81.3	0.00%	61.1	61.1	0.00%	62.7	62.7	0.00%
-645	-	-	-	143	143	0.00%	65.2	65.2	0.00%
-671	478.2	478.5	0.06%	215.1	215.4	0.14%	475.5	475.9	0.40%
633 - 632	81.3	81.3	0.00%	61.1	61.1	0.00%	62.7	62.7	0.00%
-634	81.3	81.3	0.00%	61.1	61.1	0.00%	62.7	62.7	0.00%
634 - 633	704.8	704.8	0.00%	529.7	529.7	0.00%	543.5	543.5	0.00%
645 - 632	-	-	-	143	143	0.00%	65.2	65.2	0.00%
-646	-	-	-	65.2	65.2	0.00%	65.2	65.2	0.00%
646 - 645	-	-	-	65.2	65.2	0.00%	65.2	65.2	0.00%
652 - 684	63	63.2	0.32%	-	-	-	-	-	-
671 - 632	470.2	470.2	0.00%	186.4	186.4	0.00%	420.6	420.6	0.00%
-680	0	0	0.00%	0	0	-	0	0	0.00%
-684	63	63.4	0.63%	-	-	-	71.2	71.3	0.10%
-692	229.1	229.1	0.00%	69.6	69.6	0.00%	178.4	178.4	0.00%
675 - 692	205.4	205.4	0.00%	69.6	69.6	0.00%	124.1	124.1	0.00%
680 - 671	0	0	0.00%	0	0	0.00%	0	0	0.00%
684 - 671	63	63	0.00%	-	-	-	71.2	71.3	0.10%
-611	-	-	-	-	-	-	71.2	71.3	0.10%
-652	63	63.4	0.63%	-	-	-	-	-	-
692 - 671	229.1	229.1	0.00%	69.6	69.6	0.00%	178.4	178.4	0.00%
-675	205.3	205.3	0.00%	69.6	69.6	0.00%	124.1	124.1	0.00%

Table 4.3: Validation results for a comparison of the current flow given for the IEEE 13 node radial distribution test feeder and the current flow between the nodes in the VHDL-AMS system. The difference between the two is calculated as a percentage.

4.1.3 EV Chargers on the 13 Node System

The system response of most interest is the one where many chargers are connected to a distribution system and are set to various charging states such that the aggregate effect on the system can be observed. For the 13 node system, as seen in Figure 4.7, the constant power load at node 645 was incrementally replaced by EV charger loads. Each time the percentage of loading from the EV charger was increased the current at the 645, 646, and 633 nodes was recorded. These nodes include the node at the chargers, one node away from the chargers, and two nodes away from the chargers, respectively. An FFT analysis was done on each of these waveforms in order to assess the harmonic content of each. Table 4.4 shows the TDD as calculated for the system at each of the nodes.

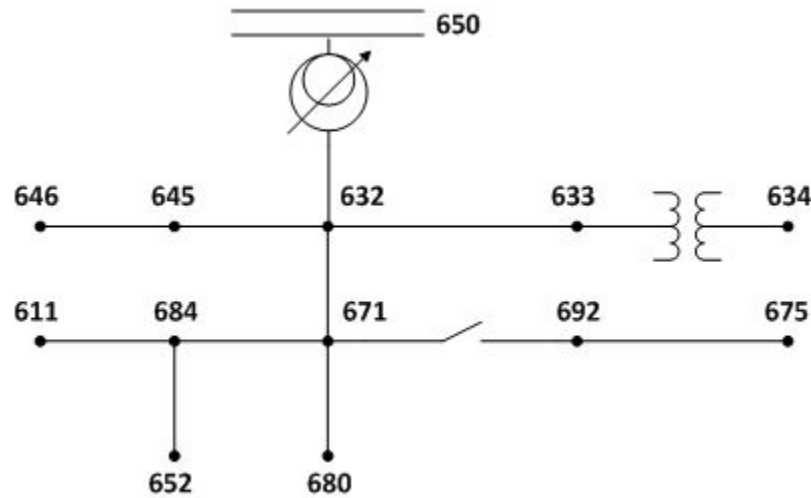


Figure 4.7: One-line diagram of IEEE 13 node system, with nodes labeled.

The current distortion limits allowed as defined by IEEE 519-1992 for the system are based on the ratio of the available short circuit current to the maximum demand load current.

TDD			
EV % of Load	Node 645	Node 646	Node 633
10	0%	0%	0%
20	0%	0%	0%
30	1%	0%	0%
40	1%	1%	0%
50	2%	1%	0%
60	3%	2%	1%
70	4%	3%	1%
80	5%	3%	1%
90	5%	4%	2%
100	6%	5%	2%

Table 4.4: TDD for three nodes in the IEEE 13 node system when EV charger loads are incrementally added at node 645.

Since the short circuit current for this system is so high compared to the maximum demand load current, current distortion levels of up to 10% are allowed through the 11th harmonic.

5 Web-Based Tool

The inspiration for this work was to create a tool that could be used to inform distribution engineers as they plan for EV growth within their service territories. The modeling tool can be used to construct and simulate proposed EV charger installations in order to plan feeder upgrades by determining the number of EV charging stations that can be added to an existing feeder or to simulate proposed feeders that will need to accommodate high EV penetration. By allowing designers to accurately predict possible side effects that EV charger controllers have on distribution assets, the tool can be used to select appropriate conductor sizing, protection settings, and transformer ratings in order to plan system upgrades, manage valuable assets, and meet TDD specifications.

The tool is available as part of a fully collaborative environment on the systemvision.com web site. The information provided in this section is intended to be provided as a stand-alone user's manual for the online tool. A set of components, including the EV charger and three-phase power models, has been uploaded to the site. At this time the components can be found at <http://app.systemvision.com/App/Shared/?sharedDesignID=773183e3-b214-4862-a415-e0736c8af2c9>, the project can be copied, and the system designer can then utilize the provided components for their design. In the final release, the models will have complete symbols attached and will be available from the within the model libraries on the site.

5.1 Components

The following sections describe the models that are available on the site as well as the parameters for the components that are user defined. How the models are used within the simulation environment as well as why the user defined parameters are of particular interest is also described.

5.1.1 Level 2 Charger

The Level 2 charger model can be installed in a schematic in either a one-phase or two-phase configuration. In either case, the p2 terminal side of the device should be grounded. The generic for "load_state" can be set at an integer value of 1 through 5, each of which represents a state in the charge cycle that is farther in the charge cycle than the integer before. With each progressive charge state the current drawn by the charger decreases and the THD value of the current increases. Different charge states can be chosen for the chargers throughout a schematic in order to simulate different EVs on the same feeder at various states of charge at the same time. The generics for voltage and frequency should not be adjusted as these are the settings at which the harmonics values for the model were calibrated. In order to change the voltage or frequency that is produced on the system, the change should be made in the generics of the power source used in the schematic.

5.1.2 Level 3 Charger

The Level 3 charger model should be instantiated in a schematic connected to a three-phase power supply and in a grounded-wye configuration. Three instances of the component are required in order to represent the full three-phase configuration of a Level 3 charger. The p2 terminal sides of the three instances should be tied together and grounded. Again, the generics for "load_state" should be set to an integer value of 1 through 5 in order to represent the state in the charge cycle that is being analyzed. It is important the all three instances of the component be set to the same state of charge, since they are representing one charger, or erroneous results will be produced. Multiple Level 3 chargers can be used on one schematic and there can also be Level 2 chargers on the same schematic. Again, the generics for voltage and frequency should not be adjusted as these are the settings at which the harmonic values for the model were calibrated and should be adjusted in the generics of the power sourced used in the schematic instead.

5.1.3 Infinite Three-Phase Source

The voltage drive source should be instantiated in the schematic by connecting it to three lines that eventually end in loads and the neutral wye pin should be connected to ground. The default generic voltage value is $227 V_{RMS}$ phase to neutral, however, this value should be adjusted according to what the phase to neutral value for the power source of the system of interest is. There are also generic options that allow the user to adjust for different amplitude and phase values for each leg of the wye. However, the default condition is balanced. Also,

the "phase_deg" generic allows the user to offset phase "A" from zero, and when this is done the other phases are adjusted by the same value, always leaving them 120° out of phase with one another. Finally, the frequency of the system can be adjusted in the generics of this component.

5.1.4 Three-Phase Transmission Line

The three-phase transmission line can be installed in the schematic when directly or indirectly connected to both a three-phase power source and three-phases of loading. The length of the line should be defined in meters as well as the self and mutual inductances and resistance per meter. The default of the line is balanced, but offsets for the self and mutual inductances and resistance can be defined.

5.1.5 Delta-Wye Transformer

The delta-wye transformer can be installed in the schematic when directly or indirectly connected to both a three-phase power source and three phases of loading. The wye-side neutral may be connected to ground. The number of turns on both the delta and wye sides can be defined independently, representing the turns ratio. The sign for the direction of the power flow can be changed with the generic "sign_power_flow" by assuming 1, the default, as "in for delta and out for wye" and -1 as "in for wye and out for delta". This model has perfect coupling so it behaves as a perfect transformer at all frequencies and has no inductance effect. The user should note that at least a minimal resistance needs to be defined in the delta winding set in order to prevent large circulating currents that are not

representative of a real transformer. Line-to-line and line-to-neutral resistances and offset resistance can also be user defined in the model's generics.

5.1.6 Delta-Delta Transformer

The delta-delta transformer can also be installed in the schematic when directly or indirectly connected to both a three-phase power source and three phases of loading. The number of turns on both sides can be defined independently, representing the turns ratio of the device. This model has perfect coupling so it behaves as a perfect transformer at all frequencies and has no inductance effect. The user should note that at least a minimal resistance needs to be defined in both winding sets in order to prevent large circulating currents that are not representative of a real transformer. Line-to-line and line-to-neutral resistances and offset resistance can also be user defined in the model's generics.

5.1.7 Single-Phase Center-Tapped Transformer

The single-phase, center-tapped transformer can be installed in a schematic connected to a single-phase of a source and grounded, on the primary side, and connected to two single-phase loads and grounded on the secondary side. The turns ratio is set using the generics "Nturns_1", "Nturns_2a_base", and "offset_Nturns_2b" in order to define the primary winding turns, the "a" split of the secondary winding turns, and, if the secondary side is uneven, the "b" split of the secondary winding turns, respectively. The resistance for each of the windings can be set in the generics as well. This model calculates the effective power dissipated and transferred internally.

5.1.8 Cable

The cable model should be inserted into the schematic in places where lengths of cable exist in the physical system. The model uses wire resistance data from the American wire gauge (AWG) table and the user should select an AWG standard integer value from 0 to 40 for the wire size as well as entering the length, in meters, of the wire. The model uses these values to internally calculate the voltage drop across it.

5.2 Capabilities

In order to simulate a given system, the user will lay out the components within the systemvision.com environment and "net" them together with wires, accordingly. Each of the EV chargers in the system will need to be set to their charging stages. It is suggested that the user try different combinations of inputs, run each simulation, and record the results in order to explore all of the different possible scenarios for a given design and draw useful conclusions about the aggregate loading on the system.

Once each simulation is run the user can probe the components and lines in the schematic in order to analyze the system response. Outputs such as voltage, current, and power flow are available and these quantities across the terminals of each component can be examined. The overall efficiency of the system, and where losses are occurring, can easily be seen. In order to get an accurate prediction of TDD, the size and length of the cables from the point of interconnection to the chargers must be defined as well as a value for the predicted average peak demand for the system over a 12 month time period. This average peak demand can be

forecast based on the number of chargers that will be installed at the site being modeled.

With this information the system designer can make final decisions regarding the conductor sizing.

6 Discussion

The results of the voltages, phase angles, and power flows of the VHDL-AMS model as compared to the IEEE 13 node radial test distribution system correlated within less than a 1%. This indicates proper construction of the model components and the feeder schematic. Deviations in all three did occur more often in the nodes that were farther away from the substation in the model. This is most likely a purely calculational error. As the length of the lines increases the magnitude of the resistances and impedances grows larger and any rounding that was used to enter their values will start to effect the final calculation of power flow on the system.

The results for the power flow analysis on the 13 node system when EV chargers were connected to the system were not as anticipated. Again, there were deviations that occurred more often at the nodes that were farther away from the substation, but any effects that the chargers caused seemed to be localized to the immediate area where they were installed. The 13 node test system is a robust system with the allowed TDD up to 10%. The impacts of the EV chargers on a smaller and less robust distribution system would likely be much greater.

The web tool offers many advantages for the user who doesn't want to learn a new coding language and complicated software in order to design and simulate new systems and components. However, because of its simplicity, it has limitations on its usefulness.

Having to set each of the charger states before every run is a onerous and time-consuming task. It would be most useful to have the simulation environment iterate through the possible combination settings for the EV chargers and tabulate the data of interest for the user. Also, it would be more convenient and less error prone for the three-phase charger to be packaged in a single symbol and a code model. Last, the FFT functionality is not yet available in the web-based environment. When this feature is added it will offer more insight into the harmonic content that is truly being contributed by the EV chargers to a given system.

7 Conclusion

Presented is a solution for modeling time-variant, non-linear EV charger loads. These models were created based on actual data from EV chargers at public charging stations. The IEEE 13 node radial test distribution feeder was used to validate the VHDL-AMS simulation environment within which the models were created. Further, the EV charger models were simulated within the 13 node test system in order to analyze their effects. Also, a web-based tool for distribution system designers was created and presented here. This tool enables designers to run simulations that include the EV charger models in order to analyze the effects that they may have on new or existing distribution systems.

The rate at which the installation of new EV chargers on power distribution networks is occurring will continue to increase into the foreseeable future. Having validated models of these EV chargers will help distribution designers to reliably predict the impacts that these loads may have on costly assets. When designers are able to make these predictions during the early planning phases of projects they are more able to mitigate the negative impacts that these EV chargers have on power systems. When these impacts are minimized, the adoption of EV chargers is made more readily by utilities and other power service providers, thereby facilitating the deployment of these pieces of equipment, which are so vital to a low carbon transportation future.

Bibliography

- [1] IEEE recommended practices and requirements for harmonic control in electrical power systems. *IEEE Std 519-1992*, 1993.
- [2] IEEE recommended practice for establishing liquid-filled and dry-type power and distribution transformer capability when supplying nonsinusoidal load currents. *IEEE Std C57.110-2008 (Revision of IEEE Std C57.110-1998)*, pages 1–52, Aug 2008.
- [3] IEEE standard definitions for the measurement of electric power quantities under sinusoidal, nonsinusoidal, balanced, or unbalanced conditions. *IEEE Std 1459-2010 (Revision of IEEE Std 1459-2000)*, pages 1–50, 2010.
- [4] P.J. Ashenden, G.D. Peterson, and D.A. Teegarden. *The System Designer's Guide to VHDL-AMS: Analog, Mixed-Signal, and Mixed-Technology Modeling*. Systems on Silicon. Elsevier Science, 2002.
- [5] T. Bohn and H. Chaudhry. Overview of SAE standards for plug-in electric vehicle. In *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, pages 1–7, 2012.
- [6] A.J. Collin, S.Z. Djokic, H.F. Thomas, and J. Meyer. Modelling of electric vehicle chargers for power system analysis. In *Electrical Power Quality and Utilisation (EPQU), 2011 11th International Conference on*, pages 1–6, Oct 2011.

- [7] P.A. Dahono, R.E. Widjaya, Syafrudin, and Qamaruzzaman. A practical approach to minimize the zero-sequence current harmonics in power distribution systems. In *Power Conversion Conference - Nagaoka 1997., Proceedings of the*, volume 2, pages 683–686 vol.2, 1997.
- [8] Li dan Chen and Yao Zhang. Model and harmonic control strategy of electric vehicle chargers connected with the power system. In *Power Engineering and Automation Conference (PEAM), 2011 IEEE*, volume 2, pages 524–528, Sept 2011.
- [9] A. Dubey, S. Santoso, and M.P. Cloud. Average-value model for plug-in hybrid electric vehicle battery chargers. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8, July 2012.
- [10] A. Elmoudi, M. Lehtonen, and H. Nordman. Effect of harmonics on transformers loss of life. In *Electrical Insulation, 2006. Conference Record of the 2006 IEEE International Symposium on*, pages 408–411, 2006.
- [11] Alexander Emanuel. *Power definitions and the physical mechanism of power flow*. Wiley, Chichester, West Sussex, 2010.
- [12] ETAP. ETAP unbalanced load flow. Web, 2005. http://etap.com/downloads/verification-validation-cases/ETAP_ComparisonResults_ublf1.pdf.
- [13] A.M.A. Haidar and K.M. Muttaqi. Behavioral characterization of electric vehicle charging loads in a distribution power grid through modeling of battery chargers. In *Industry Applications Society Annual Meeting, 2014 IEEE*, pages 1–8, Oct 2014.

- [14] Ajit K. Hiranandani. Effects of harmonics on the current carrying capacity of insulated power cables used in three phase electrical power distribution systems. In *Electricity Distribution, 2005. CIRED 2005. 18th International Conference and Exhibition on*, pages 1–5, 2005.
- [15] T. Hoevenaars, K. LeDoux, and M. Colosino. Interpreting IEEE STD 519 and meeting its harmonic limits in VFD applications. In *Petroleum and Chemical Industry Conference, 2003. Record of Conference Papers. IEEE Industry Applications Society 50th Annual*, pages 145–150, 2003.
- [16] R. Horton, J.A. Taylor, A. Maitra, and J. Halliwell. A time-domain model of a plug-in electric vehicle battery charger. In *Transmission and Distribution Conference and Exposition (T D), 2012 IEEE PES*, pages 1–5, May 2012.
- [17] I. Hunter. Power quality issues: a distribution company perspective. *Power Engineering Journal*, 15(2):75–80, 2001.
- [18] F.A. Ibarra Hernandez and C.A. Canesin. Electrical power distribution system modeling with VHDL-AMS for the construction of a real-time digital simulator using fpgas devices. In *Industry Applications (INDUSCON), 2012 10th IEEE/IAS International Conference on*, pages 1–7, Nov 2012.
- [19] W.H. Kersting. Radial distribution test feeders. *Power Systems, IEEE Transactions on*, 6(3):975–985, Aug 1991.

- [20] W.H. Kersting. Radial distribution test feeders. In *Power Engineering Society Winter Meeting, 2001. IEEE*, volume 2, pages 908–912 vol.2, 2001.
- [21] T.E. McDermott, R. Juchem, and D. Devarajan. Distribution feeder and induction motor modeling with VHDL-AMS. In *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, pages 141–146, May 2006.
- [22] F. Moller, J. Meyer, and P. Schegner. Load model of electric vehicles chargers for load flow and unbalance studies. In *Electric Power Quality and Supply Reliability Conference (PQ), 2014*, pages 9–14, June 2014.
- [23] U.S. Department of Energy. Electric vehicle charging station locations. Web, April 2015. http://www.afdc.energy.gov/fuels/electricity_locations.html.
- [24] G.A. Putrus, P. Suwanapingkarl, D. Johnston, E.C. Bentley, and M. Narayana. Impact of electric vehicles on power distribution networks. In *Vehicle Power and Propulsion Conference, 2009. VPPC '09. IEEE*, pages 827–831, 2009.
- [25] D.M. Said and K.M. Nor. Effects of harmonics on distribution transformers. In *Power Engineering Conference, 2008. AUPEC '08. Australasian Universities*, pages 1–5, 2008.
- [26] J.M. Sexauer, K.D. McBee, and K.A. Bloch. Applications of probability model to analyze the effects of electric vehicle chargers on distribution transformers. *Power Systems, IEEE Transactions on*, 28(2):847–854, May 2013.

- [27] IEEE Power & Energy Society. IEEE PES distribution system analysis subcommittee's distribution test feeder working group, August 2013.
- [28] Ashlee Vance. Putting all 20,000 electric car charging stations on the map. *Bloomberg-Businessweek Technology*, Oct 2013.
- [29] Power Quality World. Effects of harmonics on power cables. Web, September 2013. www.powerqualityworld.com/2011/08/effects-harmonics-power-cables.html.
- [30] M. Yilmaz and P.T. Krein. Review of charging power levels and infrastructure for plug-in electric and hybrid vehicles. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–8, 2012.
- [31] Shuang Zhao, S. Haghbin, O. Wallmark, M. Leksell, S. Lundmark, and O. Carlson. Transient modeling of an integrated charger for a plug-in hybrid electric vehicle. In *Power Electronics and Applications (EPE 2011), Proceedings of the 2011-14th European Conference on*, pages 1–10, Aug 2011.

Appendix A: Figures

A.1 Level 3 Charger Schematic

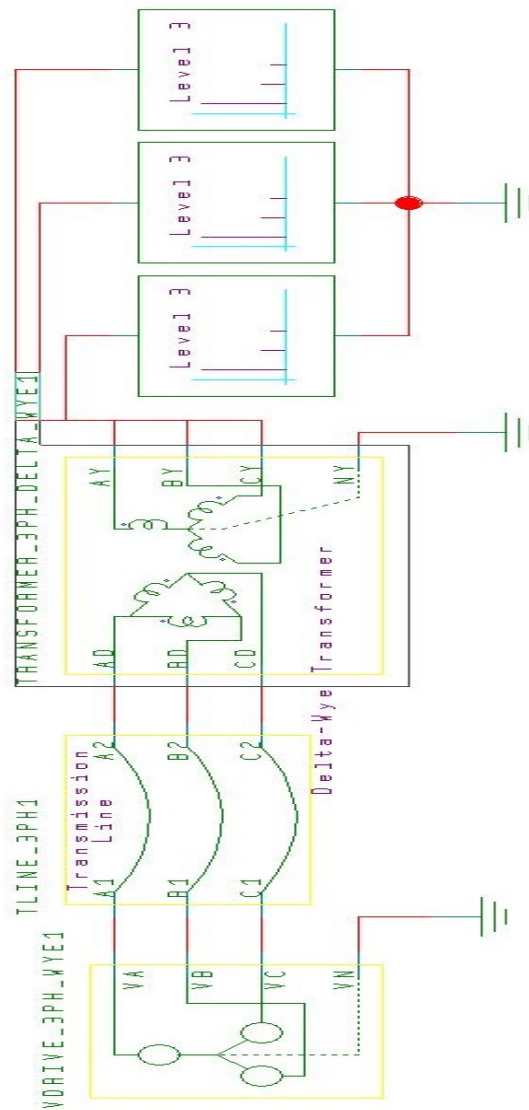


Figure A.1: Schematic of a simple Level III model.

Appendix B: VHDL-AMS Code

B.1 Level 2 Charger

```
library IEEE;

use IEEE.electrical_systems.all;

use IEEE.math_real.all;

entity level_II is

    generic (Vrms          : voltage := 120.0; --- AC Input Line
             ↪ Voltage (RMS) that was used for calibration [V]

             freq          : real    := 60.0;  --- AC Input Line
             ↪ Frequency that was used for calibration [Hz]

             load_state : integer := 1);      --- Selector for
             ↪ load state [no units]

    port (terminal p1, p2 : electrical);

begin
```

```

assert freq > 0.0

    report "freq must be greater than 0.0"

    severity error;

end entity level_II;

architecture ideal of level_II is

    quantity v across i through p1 to p2;

    quantity vnorm : real := 0.0;

    quantity vnorm_integ, vnorm_integ_dot, vnorm_dot : real :=
        ↪ 0.0;

    quantity vnorm3 : real := 0.0;

    quantity vnorm3_integ, vnorm3_integ_dot, vnorm3_dot : real
        ↪ := 0.0;

    quantity vnorm5 : real := 0.0;

    quantity vnorm5_integ, vnorm5_integ_dot, vnorm5_dot : real
        ↪ := 0.0;

```

```

constant f_pole_lpf_Hz : real := 1000.0*freq; --- Low-pass
    ↪ filter pole frequency

constant tau_lpf : real := 1.0/(math_2_pi*f_pole_lpf_Hz);

constant f_pole_hpf_Hz : real := 0.003*freq; --- Pre- and
    ↪ post- integrator High-pass filter pole frequency

constant tau_hpf : real := 1.0/(math_2_pi*f_pole_hpf_Hz);

constant w : real := math_2_pi*freq; --- Frequency in
    ↪ rad/sec

constant vpeak : voltage := Vrms*SQRT(2.0);

type load_state_array is array (1 to 10) of real_vector(0
    ↪ to 2);

-- The "Load State" current harmonic magnitude and phase
    ↪ data in the Table below.

-- Odd rows are the magnitude data, even rows are the
    ↪ corresponding phase data.

--

-- Note that the sign of the phase is interpreted such
    ↪ that a positive value means

```



```

-- the current "leads" the applied voltage. The magnitude
  ↳ and phase values can be
-- obtained directly from an FFT of any circuit's time
  ↳ domain load current waveform.
--
-- Harmonic:      1st          3rd          5th
constant load_data : load_state_array :=
(( 11.00737269,    0.1983,    0.15412),  -- Load State 1
  ↳ Magnitude (3%)
(-28.02546143,   -95.416,    81.622),  -- Load State 1
  ↳ Phase
( 11.09749781, 0.25585,  0.256703325),  -- Load State 2
  ↳ Magnitude (4%)
(32.27982644,   126.21,    11.14158045),  -- Load State
  ↳ 2 Phase
( 18.38774856,   1.393736703,    0.535130326),  -- Load
  ↳ State 3 Magnitude (8%)
(34.8613446,   -46.19719546,    8.228697006),  -- Load
  ↳ State 3 Phase
( 3.348407797, 0.23704, 0.10443),  -- Load State 4
  ↳ Magnitude (10%)

```

```

(-15.2937504, -91.928, 145.71),    -- Load State 4 Phase
(0.509127306, 0.013384, 0.046344),  -- Load State 5
↪ Magnitude (12.5%)
(78.336, 159.02, 112.21)); -- Load State 5 Phase

```

```

function calc_k_values (magdata, phasedata : in
↪ real_vector; w : in real)

return real_vector is

variable k_output : real_vector(0 to 8) := (others =>
↪ 0.0);

variable c1, c3, c5, s1, s3, s5 : real := 0.0;

variable k1i, k1, k1d1, k3i, k3, k3d1, k5i, k5, k5d1 :
↪ real := 0.0;

variable test : real := 0.0;

begin

```

```

    c1 := magdata(0)*Cos(phasedata(0)*math_pi/180.0);

    c3 := magdata(1)*Cos(phasedata(1)*math_pi/180.0);

    c5 := magdata(2)*Cos(phasedata(2)*math_pi/180.0);

```

```

s1 := magdata(0)*Sin(-1.0*phasedata(0)*math_pi/180.0);

s3 := magdata(1)*Sin(-1.0*phasedata(1)*math_pi/180.0);

s5 := magdata(2)*Sin(-1.0*phasedata(2)*math_pi/180.0);


test := c5;

k5    := 16.0*test;


test := c3 - 5.0*k5/16.0;

k3    := 4.0*test;


test := c1 - 3.0*k3/4.0 - 5.0*k5/8.0;

k1    := test;


test := s5;

if test > 0.0 then

    k5i := 80.0*w*test;

else

    k5d1 := -16.0*test/(5.0*w);

end if;

```

```

test := s3 - 5.0*k5i/(48.0*w) + 15.0*k5d1*w/16.0;

if test > 0.0 then

    k3i := 12.0*w*test;

else

    k3d1 := -4.0*test/(3.0*w);

end if;

test := s1 - 3.0*k3i/(4.0*w) - 5.0*k5i/(8.0*w) +
    ↪ 3.0*k3d1*w/4.0 + 5.0*k5d1*w/8.0;

if test > 0.0 then

    k1i := w*test;

else

    k1d1 := -1.0*test/w;

end if;

k_output := (k1i, k1, k1d1, k3i, k3, k3d1, k5i, k5,
    ↪ k5d1);

return k_output;

end function calc_k_values;

```

```

constant mag      : real_vector := load_data(2*load_state -
    ↪ 1);

constant phase : real_vector := load_data(2*load_state);

constant k : real_vector(0 to 8) :=
    ↪ calc_k_values(mag,phase,w);

constant k1i  : real := k(0);

constant k1   : real := k(1);

constant k1d1 : real := k(2);

constant k3i  : real := k(3);

constant k3   : real := k(4);

constant k3d1 : real := k(5);

constant k5i  : real := k(6);

constant k5   : real := k(7);

constant k5d1 : real := k(8);

```

begin

```

vnorm == v/vpeak;  -- Input voltage normalized to the peak
    ↪ value

vnorm3 == vnorm*vnorm*vnorm;

vnorm5 == vnorm*vnorm*vnorm*vnorm*vnorm;

```

```

vnorm_dot == vnorm'dot - tau_lpf*vnorm_dot'dot;

vnorm_integ_dot == vnorm_integ'dot -
    ⇨ tau_lpf*vnorm_integ_dot'dot;

vnorm_integ == tau_hpf*(tau_hpf*vnorm_dot -
    ⇨ 2.0*vnorm_integ_dot - tau_hpf*vnorm_integ_dot'dot);

vnorm3_dot == vnorm3'dot - tau_lpf*vnorm3_dot'dot;

vnorm3_integ_dot == vnorm3_integ'dot -
    ⇨ tau_lpf*vnorm3_integ_dot'dot;

vnorm3_integ == tau_hpf*(tau_hpf*vnorm3_dot -
    ⇨ 2.0*vnorm3_integ_dot - tau_hpf*vnorm3_integ_dot'dot);

vnorm5_dot == vnorm5'dot - tau_lpf*vnorm5_dot'dot;

vnorm5_integ_dot == vnorm5_integ'dot -
    ⇨ tau_lpf*vnorm5_integ_dot'dot;

vnorm5_integ == tau_hpf*(tau_hpf*vnorm5_dot -
    ⇨ 2.0*vnorm5_integ_dot - tau_hpf*vnorm5_integ_dot'dot);

i == kli*vnorm_integ + kl*vnorm + kld1*vnorm_dot +
    k3i*vnorm3_integ + k3*vnorm3 + k3d1*vnorm3_dot +

```

```
k5i*vnorm5_integ + k5*vnorm5 + k5d1*vnorm5_dot;
```

```
end architecture ideal;
```

B.2 Level 3 Charger

```
library IEEE;

use IEEE.electrical_systems.all;

use IEEE.math_real.all;

entity level_III is

    generic (Vrms          : voltage := 120.0; --- AC Input Line
             ↪ Voltage (RMS) that was used for calibration [V]

             freq          : real     := 60.0;  --- AC Input Line
             ↪ Frequency that was used for calibration [Hz]

             load_state : integer := 1);      --- Selector for
             ↪ load state [no units]

    port (terminal p1, p2 : electrical);

begin

    assert freq > 0.0

        report "freq must be greater than 0.0"

        severity error;
```



```
end entity level_III;
```

```
architecture ideal of level_III is
```

```
    quantity v across i through p1 to p2;
```

```
    quantity vnorm : real := 0.0;
```

```
    quantity vnorm_integ, vnorm_integ_dot, vnorm_dot : real :=  
        ↪ 0.0;
```

```
    quantity vnorm3 : real := 0.0;
```

```
    quantity vnorm3_integ, vnorm3_integ_dot, vnorm3_dot : real  
        ↪ := 0.0;
```

```
    quantity vnorm5 : real := 0.0;
```

```
    quantity vnorm5_integ, vnorm5_integ_dot, vnorm5_dot : real  
        ↪ := 0.0;
```

```
    constant f_pole_lpf_Hz : real := 1000.0*freq; --- Low-pass  
        ↪ filter pole frequency
```

```
    constant tau_lpf : real := 1.0/(math_2_pi*f_pole_lpf_Hz);
```

```

constant f_pole_hpf_Hz : real := 0.003*freq; --- Pre- and
    ↪ post- integrator High-pass filter pole frequency

constant tau_hpf : real := 1.0/(math_2_pi*f_pole_hpf_Hz);

constant w : real := math_2_pi*freq; --- Frequency in
    ↪ rad/sec

constant vpeak : voltage := Vrms*SQRT(2.0);

type load_state_array is array (1 to 10) of real_vector(0
    ↪ to 2);

-- The "Load State" current harmonic magnitude and phase
    ↪ data in the Table below.

-- Odd rows are the magnitude data, even rows are the
    ↪ corresponding phase data.
--

-- Note that the sign of the phase is interpreted such
    ↪ that a positive value means

-- the current "leads" the applied voltage. The magnitude
    ↪ and phase values can be

-- obtained directly from an FFT of any circuit's time
    ↪ domain load current waveform.

```

```

--

-- Harmonic:      1st          3rd          5th

constant load_data : load_state_array :=
(( 205.0, 5.8, 1.0045),  -- Load State 1 Magnitude (3%)
(0.0, -59.42510518, -56.87374573),  -- Load State 1 Phase
( 213.0, 4.1109, 2.4495),  -- Load State 2 Magnitude (4%)
(0.0, 50.74025168, 42.24596074),  -- Load State 2 Phase
(115.0, 2.99, 1.725),  -- Load State 3 Magnitude (8%)
(0.0, 42.00431926, 44.44183307),  -- Load State 3 Phase
( 62.0, 2.728, 2.5172),  -- Load State 4 Magnitude (10%)
(0.0, -27.79530133, -72.51569409),  -- Load State 4 Phase
(41.0, 3.0258, 2.624),  -- Load State 5 Magnitude (12.5%)
(0.0, -16.10318453, -39.08588933)); -- Load State 5 Phase

function calc_k_values (magdata, phasedata : in
↪ real_vector; w : in real)

return real_vector is

variable k_output : real_vector(0 to 8) := (others =>
↪ 0.0);

variable c1, c3, c5, s1, s3, s5 : real := 0.0;

```

```

variable kli, k1, k1d1, k3i, k3, k3d1, k5i, k5, k5d1 :
    ⇨ real := 0.0;

variable test : real := 0.0;

begin

    c1 := magdata(0)*Cos(phasedata(0)*math_pi/180.0);
    c3 := magdata(1)*Cos(phasedata(1)*math_pi/180.0);
    c5 := magdata(2)*Cos(phasedata(2)*math_pi/180.0);

    s1 := magdata(0)*Sin(-1.0*phasedata(0)*math_pi/180.0);
    s3 := magdata(1)*Sin(-1.0*phasedata(1)*math_pi/180.0);
    s5 := magdata(2)*Sin(-1.0*phasedata(2)*math_pi/180.0);

    test := c5;

    k5    := 16.0*test;

    test := c3 - 5.0*k5/16.0;

    k3    := 4.0*test;

```

```

test := c1 - 3.0*k3/4.0 - 5.0*k5/8.0;

k1    := test;

test := s5;

if test > 0.0 then

    k5i := 80.0*w*test;

else

    k5d1 := -16.0*test/(5.0*w);

end if;

test := s3 - 5.0*k5i/(48.0*w) + 15.0*k5d1*w/16.0;

if test > 0.0 then

    k3i := 12.0*w*test;

else

    k3d1 := -4.0*test/(3.0*w);

end if;

test := s1 - 3.0*k3i/(4.0*w) - 5.0*k5i/(8.0*w) +
      ⇔ 3.0*k3d1*w/4.0 + 5.0*k5d1*w/8.0;

if test > 0.0 then

    k1i := w*test;

```

```

else

    k1d1 := -1.0*test/w;

end if;

k_output := (k1i, k1, k1d1, k3i, k3, k3d1, k5i, k5,
    ↪ k5d1);

return k_output;

end function calc_k_values;

constant mag : real_vector := load_data(2*load_state -
    ↪ 1);

constant phase : real_vector := load_data(2*load_state);

constant k : real_vector(0 to 8) :=
    ↪ calc_k_values(mag,phase,w);

constant k1i : real := k(0);

constant k1 : real := k(1);

constant k1d1 : real := k(2);

constant k3i : real := k(3);

constant k3 : real := k(4);

constant k3d1 : real := k(5);

constant k5i : real := k(6);

```

```
constant k5      : real := k(7);
```

```
constant k5d1   : real := k(8);
```

```
begin
```

```
vnorm == v/vpeak;  -- Input voltage normalized to the peak
```

```
  ⇨ value
```

```
vnorm3 == vnorm*vnorm*vnorm;
```

```
vnorm5 == vnorm*vnorm*vnorm*vnorm*vnorm;
```

```
vnorm_dot == vnorm'.dot - tau_lpf*vnorm_dot'.dot;
```

```
vnorm_integ_dot == vnorm_integ'.dot -
```

```
  ⇨ tau_lpf*vnorm_integ_dot'.dot;
```

```
vnorm_integ == tau_hpf*(tau_hpf*vnorm_dot -
```

```
  ⇨ 2.0*vnorm_integ_dot - tau_hpf*vnorm_integ_dot'.dot);
```

```
vnorm3_dot == vnorm3'.dot - tau_lpf*vnorm3_dot'.dot;
```

```
vnorm3_integ_dot == vnorm3_integ'.dot -
```

```
  ⇨ tau_lpf*vnorm3_integ_dot'.dot;
```

```
vnorm3_integ == tau_hpf*(tau_hpf*vnorm3_dot -
```

```
  ⇨ 2.0*vnorm3_integ_dot - tau_hpf*vnorm3_integ_dot'.dot);
```

```

vnorm5_dot == vnorm5'dot - tau_lpf*vnorm5_dot'dot;

vnorm5_integ_dot == vnorm5_integ'dot -
    ⇨ tau_lpf*vnorm5_integ_dot'dot;

vnorm5_integ == tau_hpf*(tau_hpf*vnorm5_dot -
    ⇨ 2.0*vnorm5_integ_dot - tau_hpf*vnorm5_integ_dot'dot);

i == kli*vnorm_integ + k1*vnorm + k1d1*vnorm_dot +
    k3i*vnorm3_integ + k3*vnorm3 + k3d1*vnorm3_dot +
    k5i*vnorm5_integ + k5*vnorm5 + k5d1*vnorm5_dot;

end architecture ideal;

```


B.3 Infinite Three-Phase Source

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

use IEEE.MATH_REAL.all;

use IEEE.energy_SYSTEMS.all;

use IEEE.mechanical_systems.all;

entity vdrive_3ph_wye is

    generic (Vrms_a_pn_base : voltage      := 277.0;    --- RMS
        ↪ "A" phase to neutral voltage, and base voltage for the
        ↪ other phases [V]

        offset_V_b      : voltage      := 0.0;        ---
        ↪ Offset RMS voltage for the "B" phase, from
        ↪ Vrms_a_pn_base (can be positive or negative)
        ↪ [V]

        offset_V_c      : voltage      := 0.0;        ---
        ↪ Offset RMS voltage for the "C" phase, from
        ↪ Vrms_a_pn_base (can be positive or negative)
        ↪ [V]
```

```

Res_a_base      : resistance := 0.0;      --- "A"
    ↪ phase source resistance, base resistance for
    ↪ the other phases [Ohm]

offset_Res_b    : resistance := 0.0;      ---
    ↪ Offset source resistance for the "B" phase,
    ↪ from Res_a_base (can be positive or negative)
    ↪ [Ohm]

offset_Res_c    : resistance := 0.0;      ---
    ↪ Offset source resistance for the "C" phase,
    ↪ from Res_a_base (can be positive or negative)
    ↪ [Ohm]

freq            : real           := 60.0;  ---
    ↪ Output frequency [Hz]

phase_deg       : real           := 0.0;  ---
    ↪ Offset for phase "a" from 0.0, other phases
    ↪ shifted by same amount [degrees]

t_ramp_on       : real           := 1.0e-3); --- Time
    ↪ to ramp up each of the phases, from 0.0V
    ↪ initial value. [sec]

```

```

port (terminal va, vb, vc, vn : ELECTRICAL);

```

```
end entity vdrive_3ph_wye;
```

```
architecture default of vdrive_3ph_wye is
```

```
function calc_step_limit(freq : real) return real is
```

```
    variable lim : real;
```

```
begin
```

```
    if freq = 0.0 then
```

```
        lim := 1.0e12; -- A large value
```

```
    elsif freq < 0.0 then
```

```
        lim := 1.0 / (-20.0 * freq);
```

```
    else
```

```
        lim := 1.0 / (20.0 * freq);
```

```
    end if;
```

```
    return lim;
```

```
end function;
```

```
constant v_step_limit : real := calc_step_limit(freq);
```

```

quantity van across ian through va to vn;

limit van : voltage with v_step_limit;  -- Other phase
    ⇨ voltages will also be limited by this time step
    ⇨ control

quantity vbn across ibn through vb to vn;

quantity vcn across icn through vc to vn;


constant pi2    : real := 2.0*math_pi;

constant pi2_3  : real := 2.0*math_pi/3.0;

constant sqrt_two : real := SQRT(2.0);


constant V_a : voltage := sqrt_two*Vrms_a_pn_base;

constant V_b : voltage := sqrt_two*(Vrms_a_pn_base +
    ⇨ offset_V_b);

constant V_c : voltage := sqrt_two*(Vrms_a_pn_base +
    ⇨ offset_V_c);


constant Res_b    : resistance := Res_a_base +
    ⇨ offset_Res_b;

constant Res_c    : resistance := Res_a_base +
    ⇨ offset_Res_c;

```

```

constant w          : angular_velocity := pi2*freq;    --
    ↪ Frequency [rad/sec]

constant phase_rad : real := phase_deg*math_pi/180.0; --
    ↪ Phase "a" offset from 0.0 [rad]

quantity theta : ANGLE := 0.0;

signal start_up_sig : real := 0.0;
quantity start_up : real := 0.0;

quantity power_out_a, power_out_b, power_out_c,
    ↪ power_out_total : power := 0.0;  --- Output power in
    ↪ each phase, and total [Watt]

begin

start_up_process : process is

begin

    wait until domain = time_domain;

    start_up_sig <= 1.0;

```

```

        wait;

end process start_up_process;

theta == w*NOW + phase_rad;

start_up == start_up_sig' ramp(t_ramp_on);

van == start_up*V_a*cos(theta) + Res_a_base*ian;

vbn == start_up*V_b*cos(theta - pi2_3) + Res_b*ibn;

vcn == start_up*V_c*cos(theta + pi2_3) + Res_c*icn;

power_out_a == -1.0*van*ian;

power_out_b == -1.0*vbn*ibn;

power_out_c == -1.0*vcn*icn;

power_out_total == power_out_a + power_out_b +
    ↪ power_out_c;

end architecture default;

```

B.4 Three-Phase Transmission Line

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

use IEEE.energy_SYSTEMS.all;

entity transmission_line_3ph is

    generic (length          : real          := 100.0; --
        ↪ Length of the transmission line [meters]

        Ls_pm_a_base        : inductance    := 1.0e-6; -- Self
        ↪ inductance per meter of phase a line, and
        ↪ base inductance per meter for the other
        ↪ phases [H]

        Lm_pm_ab_base       : inductance    := 0.5e-6; --
        ↪ Mutual inductance per meter between phases a
        ↪ and b lines, and the base mutual inductance
        ↪ per meter between the other phase pairs [H]

        Res_pm_a_base       : resistance    := 1.0e-3; --
        ↪ Resistance per meter for the phase a line,
        ↪ and base resistance per meter for the other
        ↪ phases [Ohm]
```

```

offset_Ls_pm_b : inductance := 0.0;    --
    ↪ Offset self inductance per meter for the
    ↪ phase b line, from Ls_pm_a_base (can be
    ↪ positive or negative) [H]

offset_Ls_pm_c : inductance := 0.0;    --
    ↪ Offset self inductance per meter for the
    ↪ phase c line, from Ls_pm_a_base (can be
    ↪ positive or negative) [H]

offset_Lm_pm_bc : inductance := 0.0;    --
    ↪ Offset mutual inductance per meter between
    ↪ phase b and c lines, from Lm_pm_ab_base (can
    ↪ be positive or negative) [H]

offset_Lm_pm_ca : inductance := 0.0;    --
    ↪ Offset mutual inductance per meter between
    ↪ phase c and a lines, from Lm_pm_ab_base (can
    ↪ be positive or negative) [H]

offset_Res_pm_b : resistance := 0.0;    --
    ↪ Offset resistance per meter for the phase b
    ↪ line, from Res_pm_a_base (can be positive or
    ↪ negative) [Ohm]

```



```

offset_Res_pm_c : resistance := 0.0);    --
    ↪ Offset resistance per meter for the phase c
    ↪ line, from Res_pm_a_base (can be positive or
    ↪ negative) [Ohm]

port (terminal a1, b1, c1, a2, b2, c2 : ELECTRICAL);

end entity transmission_line_3ph;

architecture default of transmission_line_3ph is

    quantity va across ia through a1 to a2;
    quantity vb across ib through b1 to b2;
    quantity vc across ic through c1 to c2;

    constant Ls_a  : inductance := length*Ls_pm_a_base;
    constant Ls_b  : inductance := length*(Ls_pm_a_base +
    ↪ offset_Ls_pm_b);
    constant Ls_c  : inductance := length*(Ls_pm_a_base +
    ↪ offset_Ls_pm_c);

```

```

constant Lm_ab : inductance := length*Lm_pm_ab_base;

constant Lm_ac : inductance := length*(Lm_pm_ab_base +
    ↪ offset_Lm_pm_ca);

constant Lm_bc : inductance := length*(Lm_pm_ab_base +
    ↪ offset_Lm_pm_bc);

constant Res_a : resistance := length*Res_pm_a_base;

constant Res_b : resistance := length*(Res_pm_a_base +
    ↪ offset_Res_pm_b);

constant Res_c : resistance := length*(Res_pm_a_base +
    ↪ offset_Res_pm_c);

quantity va1 across a1 to electrical_ref;
quantity vb1 across b1 to electrical_ref;
quantity vc1 across c1 to electrical_ref;

quantity va2 across a2 to electrical_ref;
quantity vb2 across b2 to electrical_ref;
quantity vc2 across c2 to electrical_ref;

```

```

quantity power_in_a1, power_in_b1, power_in_c1,
    ↪ power_in1_total : power := 0.0;      --- Power into
    ↪ side "1" of the line [Watts]
quantity power_out_a2, power_out_b2, power_out_c2,
    ↪ power_out2_total : power := 0.0; --- Power out of side
    ↪ "2" of the line [Watts]
quantity power_dissipated_a, power_dissipated_b,
    ↪ power_dissipated_c, power_dissipated_total : power :=
    ↪ 0.0;  --- Power dissipated line resistances a, b, c
    ↪ and total [Watts]

```

begin

```

va == Res_a*ia + Ls_a*ia'.dot + Lm_ab*ib'.dot +
    ↪ Lm_ac*ic'.dot;
vb == Res_b*ib + Ls_b*ib'.dot + Lm_ab*ia'.dot +
    ↪ Lm_bc*ic'.dot;
vc == Res_c*ic + Ls_c*ic'.dot + Lm_ac*ia'.dot +
    ↪ Lm_bc*ib'.dot;

power_in_a1 == va*ia;

```

```

power_in_b1 == vb1*ib;

power_in_c1 == vc1*ic;

power_in1_total == power_in_a1 + power_in_b1 +
    ↪ power_in_c1;


power_out_a2 == va2*ia;

power_out_b2 == vb2*ib;

power_out_c2 == vc2*ic;

power_out2_total == power_out_a2 + power_out_b2 +
    ↪ power_out_c2;


power_dissipated_a == Res_a*ia**2;

power_dissipated_b == Res_b*ib**2;

power_dissipated_c == Res_c*ic**2;

power_dissipated_total == power_dissipated_a +
    ↪ power_dissipated_b + power_dissipated_c;

end architecture default;

```

B.5 Delta-Wye Transformer

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

use IEEE.energy_SYSTEMS.all;

entity transformer_3ph_delta_wye is

    generic (Nturns_d      : real      := 1.0;      -- Number
        ↪ of turns on the "Delta" windings

        Nturns_y          : real      := 1.0;      -- Number
        ↪ of turns on the "Wye" windings

        sign_power_flow : real      := 1.0;      -- Sign
        ↪ for the direction of power flow. Assumes
        ↪ power flow is positive for "in Delta, out
        ↪ Wye". Set to -1.0 otherwise.

        Res_abd_base     : resistance := 1.0e-6; --
        ↪ Line-to-line resistance between ad and bd,
        ↪ base resistance for the other "Delta"
        ↪ line-pairs [Ohm]
```

```

offset_Res_bcd  : resistance := 0.0;      -- Offset
↳ resistance for the bd to cd line-pair, from
↳ Res_abd_base (can be positive or negative)
↳ [Ohm]

offset_Res_cad  : resistance := 0.0;      -- Offset
↳ resistance for the cd to ad line-pair, from
↳ Res_abd_base (can be positive or negative)
↳ [Ohm]

Res_ay_base     : resistance := 0.0;      --
↳ Line-to-neutral winding resistance between ay
↳ and ny, base resistance for the other "Wye"
↳ phases [Ohm]

offset_Res_by   : resistance := 0.0;      -- Offset
↳ resistance for the by to ny winding, from
↳ Res_ay_base (can be positive or negative)
↳ [Ohm]

offset_Res_cy   : resistance := 0.0;      -- Offset
↳ resistance for the cy to ny winding, from
↳ Res_ay_base (can be positive or negative)
↳ [Ohm]

```

```

    r_leak_delta      : resistance := 1.0e6); --
    ↪ Leakage resistance from the "ad" connection
    ↪ (on the delta winding set) to ground
    ↪ (electrical_ref) [Ohm]

port (terminal ad, bd, cd,                      --- "Delta"
    ↪ windings' terminals

    ay, by, cy, ny : ELECTRICAL); --- "Wye"
    ↪ windings' terminals, with ny the
    ↪ neutral

begin

    assert r_leak_delta > 0.0
        report "r_leak_delta must be greater than 0.0"
        severity error;

end entity transformer_3ph_delta_wye;

architecture default of transformer_3ph_delta_wye is

```

```

quantity vabd across iabd through ad to bd;

quantity vbcd across ibcd through bd to cd;

quantity vcad across icad through cd to ad;


quantity vad across i_leak through ad to electrical_ref;


quantity vay across iay through ay to ny;

quantity vby across iby through by to ny;

quantity vcy across icy through cy to ny;


constant Res_bcd      : resistance := Res_abd_base +
    ↪ offset_Res_bcd;

constant Res_cad      : resistance := Res_abd_base +
    ↪ offset_Res_cad;

constant Res_by       : resistance := Res_ay_base  +
    ↪ offset_Res_by;

constant Res_cy       : resistance := Res_ay_base  +
    ↪ offset_Res_cy;


constant Nyd : real := Nturns_y/Nturns_d;

```



```

quantity vay_m, vby_m, vcy_m, vabd_m, vbcd_m, vcad_m :

  ⇨ voltage := 0.0;    --- Induced voltages in the windings

  ⇨ [V]

quantity power_abd, power_bcd, power_cad, power_d_total :

  ⇨ power := 0.0;    --- Power into or out of the "Delta"

  ⇨ windings [Watts]

quantity power_ay, power_by, power_cy, power_y_total :

  ⇨ power := 0.0;    --- Power into or out of the "Wye"

  ⇨ windings [Watts]

quantity power_dissipated_y, power_dissipated_d,

  ⇨ power_dissipated_total : power := 0.0; --- Power

  ⇨ dissipated in the "Wye" and "Delta" winding sets, and

  ⇨ the total [Watts]

```

begin

```

vay    == Res_ay_base*iay + vay_m;

vay_m == Nyd*vabd_m;

iabd   == -1.0*Nyd*iay;

```

```
vabd == Res_abd_base*iabd + vabd_m;
```

```
vby == Res_by*iby + vby_m;
```

```
vby_m == Nyd*vbcd_m;
```

```
ibcd == -1.0*Nyd*iby;
```

```
vbcd == Res_bcd*ibcd + vbcd_m;
```

```
vcy == Res_cy*icy + vcy_m;
```

```
vcy_m == Nyd*vcad_m;
```

```
icad == -1.0*Nyd*icy;
```

```
vcad == Res_cad*icad + vcad_m;
```

```
power_abd == sign_power_flow*vabd*iabd;
```

```
power_bcd == sign_power_flow*vbcd*ibcd;
```

```
power_cad == sign_power_flow*vcad*icad;
```

```
power_d_total == power_abd + power_bcd + power_cad;
```

```
power_ay == -1.0*sign_power_flow*vay*iay;
```

```
power_by == -1.0*sign_power_flow*vby*iby;
```

```
power_cy == -1.0*sign_power_flow*vcy*icy;
```

```
power_y_total == power_ay + power_by + power_cy;
```

```

power_dissipated_d == Res_abd_base*iabd**2 +
    ↪ Res_bcd*ibcd**2 + Res_cad*icad**2;

power_dissipated_y == Res_ay_base*iay**2 + Res_by*iby**2 +
    ↪ Res_cy*icy**2;

power_dissipated_total == power_dissipated_y +
    ↪ power_dissipated_d;

i_leak == vad/r_leak_delta;

end architecture default;

```

B.6 Delta-Delta Transformer

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

use IEEE.energy_SYSTEMS.all;

entity transformer_3ph_delta_delta is

    generic (Nturns_1      : real      := 1.0;      -- Number
        ↪ of turns on windings in set 1

        Nturns_2          : real      := 1.0;      -- Number
        ↪ of turns on windings in set 2

        Res_ab1_base      : resistance := 1.0e-6; --
        ↪ Line-to-line resistance between a1 and b1,
        ↪ base resistance for the other "1" line-pairs
        ↪ [Ohm]

        offset_Res_bc1    : resistance := 0.0;      -- Offset
        ↪ resistance for the b1 to c1 line-pair, from
        ↪ Res_ab1_base (can be positive or negative)
        ↪ [Ohm]
```

```

offset_Res_cal : resistance := 0.0;    -- Offset
↳ resistance for the c1 to a1 line-pair, from
↳ Res_ab1_base (can be positive or negative)
↳ [Ohm]

Res_ab2_base   : resistance := 1.0e-6; --
↳ Line-to-line resistance between a2 and b2,
↳ base resistance for the other "2" line-pairs
↳ [Ohm]

offset_Res_bc2 : resistance := 0.0;    -- Offset
↳ resistance for the b2 to c2 line-pair, from
↳ Res_ab2_base (can be positive or negative)
↳ [Ohm]

offset_Res_ca2 : resistance := 0.0;    -- Offset
↳ resistance for the c2 to a2 line-pair, from
↳ Res_ab2_base (can be positive or negative)
↳ [Ohm]

r_leak_delta   : resistance := 1.0e6); --
↳ Leakage resistance from the "a" connections
↳ (on both delta winding sets) to ground
↳ (electrical_ref) [Ohm]

```

```

    port (terminal a1, b1, c1, a2, b2, c2 : ELECTRICAL);

begin

    assert r_leak_delta > 0.0

        report "r_leak_delta must be greater than 0.0"

        severity error;

end entity transformer_3ph_delta_delta;


architecture default of transformer_3ph_delta_delta is

    quantity vab1 across iab1 through a1 to b1;
    quantity vbc1 across ibc1 through b1 to c1;
    quantity vca1 across ica1 through c1 to a1;

    quantity vab2 across iab2 through a2 to b2;
    quantity vbc2 across ibc2 through b2 to c2;
    quantity vca2 across ica2 through c2 to a2;

```

quantity va1 across i1_leak through a1 **to** electrical_ref;

quantity va2 across i2_leak through a2 **to** electrical_ref;

constant Res_bc1 : resistance := Res_ab1_base +

↪ offset_Res_bc1;

constant Res_ca1 : resistance := Res_ab1_base +

↪ offset_Res_ca1;

constant Res_bc2 : resistance := Res_ab2_base +

↪ offset_Res_bc2;

constant Res_ca2 : resistance := Res_ab2_base +

↪ offset_Res_ca2;

constant N12 : real := Nturns_1/Nturns_2;

quantity vab1_m, vbc1_m, vca1_m, vab2_m, vbc2_m, vca2_m :

↪ voltage := 0.0; --- *Induced voltages in the windings*

↪ [V]

```

quantity power_in_ab1, power_in_bc1, power_in_ca1,
    ↪ power_in1_total : power := 0.0;      --- Power into the
    ↪ side "1" windings [Watts]

quantity power_out_ab2, power_out_bc2, power_out_ca2,
    ↪ power_out2_total : power := 0.0; --- Power out of the
    ↪ side "2" windings [Watts]

quantity power_dissipated1, power_dissipated2,
    ↪ power_dissipated_total : power := 0.0; --- Power
    ↪ dissipated in winding sets 1, 2 and total [Watts]

```

begin

```

vab1    == Res_ab1_base*iab1 + vab1_m;

vab1_m == N12*vab2_m;

iab2    == -1.0*N12*iab1;

vab2    == Res_ab2_base*iab2 + vab2_m;

vbc1    == Res_bc1*ibc1 + vbc1_m;

vbc1_m == N12*vbc2_m;

ibc2    == -1.0*N12*ibc1;

vbc2    == Res_bc2*ibc2 + vbc2_m;

```



```
vca1 == Res_ca1*ica1 + vca1_m;
```

```
vca1_m == N12*vca2_m;
```

```
ica2 == -1.0*N12*ica1;
```

```
vca2 == Res_ca2*ica2 + vca2_m;
```

```
power_in_ab1 == vab1*iab1;
```

```
power_in_bc1 == vbc1*ibc1;
```

```
power_in_ca1 == vca1*ica1;
```

```
power_in1_total == power_in_ab1 + power_in_bc1 +  
    ↪ power_in_ca1;
```

```
power_out_ab2 == -1.0*vab2*iab2;
```

```
power_out_bc2 == -1.0*vbc2*ibc2;
```

```
power_out_ca2 == -1.0*vca2*ica2;
```

```
power_out2_total == power_out_ab2 + power_out_bc2 +  
    ↪ power_out_ca2;
```

```
power_dissipated1 == Res_ab1_base*iab1**2 +
```

```
    ↪ Res_bc1*ibc1**2 + Res_ca1*ica1**2;
```

```

power_dissipated2 == Res_ab2_base*iab2**2 +
    ↪ Res_bc2*ibc2**2 + Res_ca2*ica2**2;

power_dissipated_total == power_dissipated1 +
    ↪ power_dissipated2;

i1_leak == va1/r_leak_delta;

i2_leak == va2/r_leak_delta;

end architecture default;

```

B.7 Single-Phase Center Tapped Transformer

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

use IEEE.energy_SYSTEMS.all;

entity transformer_lph_centertap is

    generic (Nturns_1          : real          := 2.0;  -- Number
    ↪ of turns on winding 1 [no units]

            Nturns_2a_base    : real          := 1.0;  -- Number
    ↪ of turns on the "a" split of the secondary
    ↪ winding, used as base number for both the 2a
    ↪ and 2b splits [no units]

            offset_Nturns_2b : real          := 0.0;  -- Offset
    ↪ number of turns on winding 2b, relative to
    ↪ winding 2a (can be positive or negative) [no
    ↪ units]

            Res_1             : resistance := 0.0;  --
    ↪ Resistance of winding 1 [Ohm]
```

```

Res_2a_base      : resistance := 0.0;  --
    ↪ Resistance of the "a" split of the secondary
    ↪ winding, used as the base resistance for both
    ↪ the 2a and 2b splits [Ohms]

offset_Res_2b    : resistance := 0.0); -- Offset
    ↪ resistance of winding 2b, relative to
    ↪ Res_2a_base (can be positive or negative)
    ↪ [Ohm]

port (terminal term_1p, term_1n, term_2ap, term_2ct,
    ↪ term_2bn : ELECTRICAL);

end entity transformer_1ph_centertap;

architecture default of transformer_1ph_centertap is

    quantity v1  across i1  through term_1p to term_1n;
    quantity v2a across i2a through term_2ap to term_2ct;
    quantity v2b across i2b through term_2ct to term_2bn;

```

```

quantity v1_m, v2a_m, v2b_m : voltage := 0.0;  --- Induced
    ↪ voltages in the windings [V]

constant N12a : real := Nturns_1/Nturns_2a_base;

constant N12b : real := Nturns_1/(Nturns_2a_base +
    ↪ offset_Nturns_2b);

constant Res_2a : resistance := Res_2a_base;

constant Res_2b : resistance := Res_2a_base +
    ↪ offset_Res_2b;

quantity power_in_1 : power := 0.0;
    ↪ --- Power into the side "1" winding [Watts]

quantity power_out_2a, power_out_2b, power_out_total :
    ↪ power := 0.0; --- Power out of the side "2" windings
    ↪ [Watts]

quantity power_dissipated_1, power_dissipated_2a,
    ↪ power_dissipated_2b, power_dissipated_total : power :=
    ↪ 0.0; --- Power dissipated in windings 1, 2a, 2b and
    ↪ total [Watts]

```

begin

$v1 == Res_1 \cdot i1 + v1_m;$

$v1_m == N12a \cdot v2a_m;$

$v1_m == N12b \cdot v2b_m;$

$i1 == -1.0 \cdot i2a / N12a - i2b / N12b;$

$v2a == Res_2a \cdot i2a + v2a_m;$

$v2b == Res_2b \cdot i2b + v2b_m;$

$power_in_1 == v1 \cdot i1;$

$power_out_2a == -1.0 \cdot v2a \cdot i2a;$

$power_out_2b == -1.0 \cdot v2b \cdot i2b;$

$power_out_total == power_out_2a + power_out_2b;$

$power_dissipated_1 == Res_1 \cdot i1^2;$

$power_dissipated_2a == Res_2a \cdot i2a^2;$

$power_dissipated_2b == Res_2b \cdot i2b^2;$

$power_dissipated_total == power_dissipated_1 +$

$\rightarrow power_dissipated_2a + power_dissipated_2b;$

```
end architecture default;
```

B.8 Cable

```
library IEEE;

use IEEE.electrical_systems.all;

entity wire_R is

    generic (gauge           : integer := 10;    -- Select
             ↪ Wire Gauge - AWG Standard (Integer value, 0 to 40) [No
             ↪ Units]

             length          : real    := 1.0); -- Wire
             ↪ length [meters]

    port (terminal p, m : electrical);

begin

    assert gauge = 0 or gauge = 1 or gauge = 2 or gauge = 3 or
    ↪ gauge = 4 or gauge = 5 or gauge = 6 or gauge = 7 or
    ↪ gauge = 8 or

        gauge = 9 or gauge = 10 or gauge = 11 or gauge = 12
    ↪ or gauge = 13 or gauge = 14 or gauge = 15 or
    ↪ gauge = 16 or

        gauge = 17 or gauge = 18 or gauge = 19 or gauge =
    ↪ 20 or gauge = 21 or gauge = 22 or gauge = 23 or
    ↪ gauge = 24 or
```



```

    gauge = 25 or gauge = 26 or gauge = 27 or gauge =
    ↪ 28 or gauge = 29 or gauge = 30 or gauge = 31 or
    ↪ gauge = 32 or

    gauge = 33 or gauge = 34 or gauge = 35 or gauge =
    ↪ 36 or gauge = 37 or gauge = 38 or gauge = 39 or
    ↪ gauge = 40

    report "guage must be an integer between 0 and 40"

    severity error;

end entity wire_R;

architecture default of wire_R is

    quantity v across i through p to m;

    function wire_resistance_select (selection : in integer)
        return real is
    begin
        if selection = 0 then
            return 0.328e-3;           -- Resistance [Ohms per
            ↪ meter]

        elsif selection = 1 then
            return 0.4066e-3;

```

```
elseif selection = 2 then
    return 0.5127e-3;
elseif selection = 3 then
    return 0.6465e-3;
elseif selection = 4 then
    return 0.8152e-3;
elseif selection = 5 then
    return 1.028e-3;
elseif selection = 6 then
    return 1.296e-3;
elseif selection = 7 then
    return 1.634e-3;
elseif selection = 8 then
    return 2.061e-3;
elseif selection = 9 then
    return 2.599e-3;
elseif selection = 10 then
    return 3.277e-3;
elseif selection = 11 then
    return 4.132e-3;
elseif selection = 12 then
```

```
    return 5.211e-3;
elseif selection = 13 then
    return 6.571e-3;
elseif selection = 14 then
    return 8.286e-3;
elseif selection = 15 then
    return 10.45e-3;
elseif selection = 16 then
    return 13.17e-3;
elseif selection = 17 then
    return 16.61e-3;
elseif selection = 18 then
    return 20.95e-3;
elseif selection = 19 then
    return 26.42e-3;
elseif selection = 20 then
    return 33.31e-3;
elseif selection = 21 then
    return 42.00e-3;
elseif selection = 22 then
    return 52.96e-3;
```

```
elseif selection = 23 then
    return 66.79e-3;
elseif selection = 24 then
    return 84.22e-3;
elseif selection = 25 then
    return 0.1062;
elseif selection = 26 then
    return 0.1339;
elseif selection = 27 then
    return 0.1689;
elseif selection = 28 then
    return 0.2129;
elseif selection = 29 then
    return 0.2685;
elseif selection = 30 then
    return 0.3386;
elseif selection = 31 then
    return 0.4269;
elseif selection = 32 then
    return 0.5383;
elseif selection = 33 then
```

```

        return 0.6788;

    elsif selection = 34 then

        return 0.8560;

    elsif selection = 35 then

        return 1.079;

    elsif selection = 36 then

        return 1.361;

    elsif selection = 37 then

        return 1.716;

    elsif selection = 38 then

        return 2.164;

    elsif selection = 39 then

        return 2.729;

    elsif selection = 40 then

        return 3.441;

    end if;

end function wire_resistance_select;

constant r : real := wire_resistance_select(gauge);

begin

```

```
v == i*r*length;
```

```
end architecture default;
```

B.9 Voltage Source

```
library IEEE;

use IEEE.MATH_REAL.all;

-- Use IEEE natures and packages

use IEEE.ELECTRICAL_SYSTEMS.all;

entity v_sine is

    generic (

        freq      : real;           -- frequency [Hz]
        amplitude : voltage;        -- amplitude [V]
        phase      : real := 0.0;    -- initial phase
        ↪ [Degree]
        offset     : voltage := 0.0; -- DC value [V]
        df         : real := 0.0;    -- damping factor
        ↪ [1/s]
        ac_mag     : voltage := 0.0; -- AC magnitude [V]
        ac_phase   : real := 0.0;    -- AC phase [Degree]

    port (

        terminal pos, neg : electrical);
```

```

end entity v_sine;

-----

-- Ideal Architecture

-----

architecture ideal of v_sine is

    function calc_limit(freq : real) return real is

        variable lim : real;

    begin

        if freq = 0.0 then

            lim := 1.0e12; -- A large value

        elsif freq < 0.0 then

            lim := 1.0 / (-20.0 * freq);

        else

            lim := 1.0 / (20.0 * freq);

        end if;

        return lim;

    end function;

    constant v_limit : real := calc_limit(freq);

```



```

-- Declare Branch Quantities

quantity v across i through pos to neg;

limit v : voltage with v_limit;

-- Declare Quantity for Phase in radians (calculated below)

quantity phase_rad : real;

-- Declare Quantity in frequency domain for AC analysis

quantity ac_spec      : real spectrum ac_mag,

    ↪ math_2_pi*ac_phase/360.0;

begin

-- Convert phase to radians

phase_rad == math_2_pi *(freq * NOW + phase / 360.0);

if domain = quiescent_domain or domain = time_domain use

    v == offset + amplitude * sin(phase_rad) * EXP(-NOW *

    ↪ df);

else

    v == ac_spec;                -- used for Frequency (AC)

    ↪ analysis

end use;

```

```
end architecture ideal;
```

B.10 Single-Phase Transformer

```
library IEEE;

use ieee.math_real.all;

-- Use IEEE natures and packages

use IEEE.electrical_systems.all;

entity transformer is

    generic (

        rp : resistance;           -- Primary winding resistance
        ⇨ [Ohm]

        rs : resistance;           -- Secondary winding resistance
        ⇨ [Ohm]

        lp : inductance;           -- Primary inductance [H]

        ls : inductance;           -- Secondary inductance [H]

        k  : real := 1.0);         -- Coupling coefficient [No Units]

    port (terminal

        p1,                         -- Primary terminal
        ⇨ 1
```

```

        p2,                                -- Primary terminal
        ⇨ 2

        s1,                                -- Secondary
        ⇨ terminal 1

        s2 : electrical);                 -- Secondary
        ⇨ terminal 2

end entity transformer;

-----

-- Ideal architecture
-----

architecture ideal of transformer is

    constant m : real := k * sqrt(lp * ls); -- Mutual
    ⇨ inductance

    quantity vpri across ipri through p1 to p2;

    quantity vsec across isec through s1 to s2;

begin

    vpri == ipri * rp + lp * ipri'.dot + m * isec'.dot;

```

```
vsec == isec * rs + m * ipri'.dot + ls * isec'.dot;
```

```
end architecture ideal;
```

B.11 Voltage Regulator

```
library IEEE;

use IEEE.ELECTRICAL_SYSTEMS.all;

entity linear_reg_ideal is

    generic (v_set : voltage := 3.3;           -- Output voltage
             ↪ setpoint [V]

             vdo    : voltage := 0.5;         -- Minimum input
             ↪ voltage relative to the output voltage setpoint
             ↪ [V]

             i_max  : current := 400.0e-3);    -- Maximum output
             ↪ current [A]

    port (terminal p_in, p_out, p_gnd : electrical);

end entity linear_reg_ideal;

architecture default of linear_reg_ideal is

    quantity v_in across p_in to p_gnd;
```

```

quantity i_in_out through p_in to p_out;

quantity v_out across p_out to p_gnd;

--- Control parameters for enforcing i_max

constant r_min      : resistance := vdo/i_max;    -- Minimum
    ↪ value of variable resistance [Ohm]

constant kr_ilimit : real        := 1.0e3;      --
    ↪ Additional resistance gain factor for over-current (i
    ↪ > i_max) [Ohm/Amp]

quantity r          : resistance := r_min;        --
    ↪ Variable resistance of the regulator [Ohm]

--- Control parameters for enforcing vdo

constant v_dropout : voltage      := v_set + vdo; --
    ↪ Absolute dropout voltage [V]

constant r_vdo     : resistance := 1.0e6;        --
    ↪ Variable resistance value during vdo limiting [Ohm]

constant tr_tf_vdo : real         := 10.0e-6;    -- VDO
    ↪ turn-off (drop-out) and turn-on (recovery) time [sec]

signal sig_r_min  : resistance := r_min;        --
    ↪ Adjustable minimum of variable resistance [Ohm]

```

```

quantity dyn_r_min : resistance := r_min;           -- Dynamic
↳ (continuous) value of the minimum resistance [Ohm]

```

```

begin

```

```

dropout : process (v_in'above(v_dropout)) is

```

```

begin

```

```

    if v_in'above(v_dropout) then

```

```

        sig_r_min <= r_min;

```

```

    else

```

```

        sig_r_min <= r_vdo;

```

```

    end if;

```

```

end process dropout;

```

```

dyn_r_min == sig_r_min'ramp(tr_tf_vdo, tr_tf_vdo);

```

```

limit_r : procedural is

```

```

begin

```

```

    if i_in_out < 1.0e-9 then

```

```

        r := (v_in - v_set)/1.0e-9;

```

```

    else

```



```

        r := (v_in - v_set)/i_in_out;

    end if;

    if i_in_out > i_max then

        r := r + kr_ilimit*(i_in_out - i_max);

    end if;

    if r < dyn_r_min then

        r := dyn_r_min;

    end if;

end procedural limit_r;

i_in_out*r == v_in - v_out;

end architecture default;

```

B.12 Resistor

```
library IEEE;

use IEEE.electrical_systems.all;

entity resistor is

    generic (

        res : resistance);          -- Resistance (no initial
        ↪ value) [Ohm]

    port (

        terminal p1, p2 : electrical);

end entity resistor;

-----

-- Ideal Architecture ( $V = I \cdot R$ )

-----

architecture ideal of resistor is

    quantity v across i through p1 to p2;
```

```
begin

-- Fundamental equation

v == i*res;

end architecture ideal;
```

B.13 Inductor

```
-- Use proposed IEEE natures and packages

library IEEE;

use IEEE.electrical_systems.all;

entity inductor is

    generic (

        ind : inductance;           -- Nominal inductance [H]

        i_ic : real := real'low);   -- Initial current (use
        ⇨ IF statement below to activate) [A]

    port (

        terminal p1, p2 : electrical);

end entity inductor;

-----

-- Ideal Architecture ( $V = L * di/dt$ )

-- Includes initial condition

-----
```

```

architecture ideal of inductor is

    -- Declare Branch Quantities

    quantity v across i through p1 to p2;

begin

    if domain = quiescent_domain and i_ic /= real'low use

        i == i_ic;

    else

        v == ind * i'.dot;                                -- Fundamental
        ⇔ equation

    end use;

end architecture ideal;

```

B.14 Capacitor

```
-- Use proposed IEEE natures and packages

library IEEE;

use IEEE.electrical_systems.all;

entity capacitor is

    generic (

        cap          : capacitance;           -- Capacitance [F]

        v_ic         : real := real'low      -- Initial voltage

        ⇨ (activated by

        );                                           -- IF statement

        ⇨ below) [V]

    port (

        terminal p1, p2 : electrical);

end entity capacitor;

-----

-- Ideal Architecture  ( $I = C * dV/dt$ )
```

```

-- Includes initial condition
-----

architecture ideal of capacitor is

    quantity v across i through p1 to p2;

begin

    if domain = quiescent_domain and v_ic /= real'low use

        v == v_ic;

    else

        i == cap * v'dot;           -- Fundamental
        ⇔ equation

    end use;

end architecture ideal;

```

B.15 Constant Power Load

```
.SUBCKT CONSTPOWER N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/170000  
  
L N3 N2 L=331.565  
  
.ENDS CONSTPOWER
```

```
.SUBCKT CONST10 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/17000  
  
L N3 N2 L=10000/377  
  
.ENDS CONST10
```

```
.SUBCKT CONST38 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/66000  
  
L N3 N2 L=38000/377  
  
.ENDS CONST38
```

```
.SUBCKT CONST60 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/68000  
  
L N3 N2 L=60000/377  
  
.ENDS CONST60
```



```
.SUBCKT CONST68 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/117000  
  
L N3 N2 L=68000/377  
  
.ENDS CONST68
```

```
.SUBCKT CONST90 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/120000  
  
L N3 N2 L=90000/377  
  
.ENDS CONST90
```

```
.SUBCKT CONST110 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/160000  
  
L N3 N2 L=110000/377  
  
.ENDS CONST110
```

```
.SUBCKT CONST212 N1 N2  
  
R N1 N3 R=(V(N1)-V(N2))**2/290000  
  
L N3 N2 L=212000/377  
  
.ENDS CONST212
```

```
.SUBCKT CONST220 N1 N2
```

```
R N1 N3 R=(V(N1)-V(N2))**2/385000
```

```
L N3 N2 L=220000/377
```

```
.ENDS CONST220
```

```
.SUBCKT CONST190 N1 N2
```

```
R N1 N3 R=(V(N1)-V(N2))**2/485000
```

```
L N3 N2 L=190000/377
```

```
.ENDS CONST190
```

B.16 Constant Current Load

```
library IEEE;

use ieee.math_real.all;

-- Use IEEE natures and packages

use IEEE.electrical_systems.all;

entity ieee_i_source is

generic (

    freq : real; -- frequency [Hertz]

    amplitude : current; -- amplitude [Amps]

    phase : real := 0.0; -- initial phase [Degrees]

    offset : current := 0.0; -- DC value [Amps]

    df : real := 0.0; -- damping factor [1/second]

    ac_mag : current := 0.0; -- AC magnitude [Amps]

    ac_phase : real := 0.0); -- AC phase [Degrees]

    port (

        terminal pos: ELECTRICAL;

        terminal neg: ELECTRICAL

    );
```

```

end entity ieee_i_source;

architecture ideal of ieee_i_source is

function calc_limit(freq : real) return real is

variable lim : real;

begin

if freq = 0.0 then

    lim := 1.0e12; -- A large value

elsif freq < 0.0 then

    lim := 1.0 / (-20.0 * freq);

else

    lim := 1.0 / (20.0 * freq);

end if;

return lim;

end function;

constant i_limit : real := calc_limit(freq);

quantity v across i through pos to neg;

limit i : current with i_limit;

quantity phase_rad : real;

quantity ac_spec : real spectrum ac_mag,

math_2_pi*ac_phase/360.0;

```

```

begin
    phase_rad == math_2_pi *(freq * NOW + phase / 360.0);

    if domain = quiescent_domain or domain = time_domain

use
    i == offset + amplitude * sin(phase_rad) * exp(-NOW *
df);

    else

    i == ac_spec;

    end use;

end architecture ideal;

```